

## Adaptive video streaming in lossy networks: versions or layers?

Ivana Radulovic      Pascal Frossard  
 Signal Processing Institute  
 Swiss Federal Institute of Technology  
 Lausanne, Switzerland  
 {ivana.radulovic,pascal.frossard}@epfl.ch

Olivier Verscheure  
 IBM T.J. Watson Research Center  
 P.O.Box 704  
 Yorktown Heights, NY - USA  
 ov1@us.ibm.com

### Abstract

This paper tackles low delay adaptive video streaming over error-prone networks. Our framework consists of an encoding station, an edge server and a set of clients with various access rates. The edge server is capable of performing simple error concealment operations on the incoming data before forwarding the adapted media to its clients. We study two encoding scenarios: versions (multiple encodings at various output rates) and layers. We develop a unified end-to-end distortion model, which we use to derive the optimal coding strategy for both scenarios. Finally we analyze the performance of MPEG-4 coded versions against MPEG-4 FGS-coded layers in rate-constrained lossy environments. Experiments show that versions perform better than layers when the constraint on the aggregate rate is somewhat relaxed, for low to medium packet loss ratios.

### 1 Introduction

A growing number of enterprises use *live* streaming video to convey news clips or corporate communications to their employees or customers. Clearly, achieving efficient streaming of live video to a wide heterogeneous user population poses many technical challenges.

We consider the scenario illustrated by Figure 1. An encoding station  $E$  compresses the input video signal and *unreliably* streams it (i.e., UDP, possibly multicast) to an edge server  $S$  located at the frontier between the enterprise network (extranet) and the client population. The edge server performs simple error concealment strategies before *reliably* streaming (i.e., TCP/HTTP for security reasons) the appropriate bit rate to each of its clients.

We study and compare the following two situations: (i) The encoding station produces layered-coded video (bandwidth efficient but sensitive to packet loss). In this case

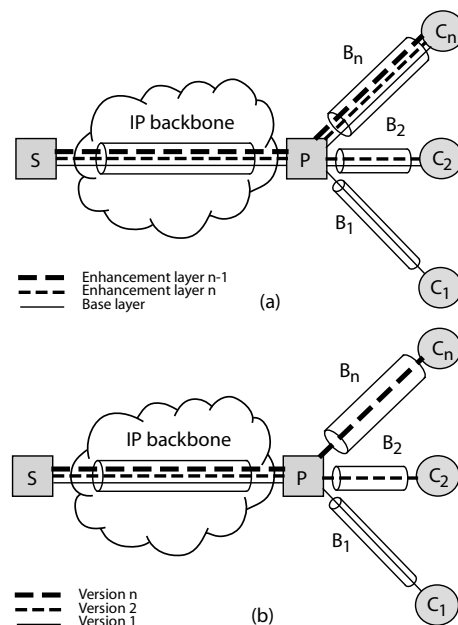


Figure 1. Adaptive streaming of (a) video versions and (b) layered video.

(see Figure 1(a)), the edge server forwards the base layer to all clients and a subset (possibly all) of the enhancement layers to clients with adequate access rates. (ii) The encoding station creates multiple versions (good error resilience but bandwidth-greedy). The edge server replaces corrupted video frames by the corresponding video frames from the immediately lower version rate such that the rate constraint is not violated and forwards the appropriate version to its clients (see Figure 1(b)). Note that in the layered-coded case there is no simple error concealment strategy the edge server can implement.

Our work builds on prior research on benchmarking versions- against layered-coded videos [2, 4], in which it was shown that versions were usually the preferred adaptive coding and streaming strategy over layers for lossless

This work has been partly supported by the Swiss National Science Foundation.

environments. We go a step further and study the impact of packet loss for low-delay adaptive video streaming. Our contribution is twofold. First, we introduce a novel rate-distortion model. Our model encompasses source coding and channel degradation for both versions- and layered-coded video and is shown to adequately fit real data. Second, we study the streaming performance of versions- against layered-coded video for the scenario described above. Versions-coded videos are shown to usually perform better under realistic assumptions.

The paper is organized as follows: Section 2 formally defines the problem as a dynamic programming problem. Our unified rate-distortion model is described and validated against real data in Section 3. The optimization problem is solved and experimental results are analysed in Section 4. Finally Section 5 provides final conclusions to this work.

## 2 Problem formulation

Let assume the heterogeneous clients  $C_i$  are sorted according to their increasing access bandwidth  $B_i$  on the last, and lossless segment of the streaming path. Let  $D_i$  denote the distortion perceived by the client  $C_i$ , and that it depends on the streaming strategy and the loss process on the Internet network. The first streaming policy (see Figure 1(a)), is based on a generic scalable encoding algorithm, which generates  $N$  layers and additive streams of rates  $R_1$  to  $R_N$  (i.e., the layer  $n$  has a rate of  $r_n = R_{n+1} - R_n$ ). The second streaming strategy, represented in Figure 1(b), builds  $N$  different versions of the media information with increasing rates  $R_1$  to  $R_N$ , and sends each version to a different client (or to a group of clients with similar access rates). We assume that the number of versions, respectively layers, is equal to the number of client clusters, that group client with similar access bandwidth. Note that the clustering algorithm is not addressed here due to space constraints, but related work may be found in [6] for interested readers. In this context, the optimization problem, similar for both streaming strategies, can be written as follows.

**Problem formulation:** Given (i) the access rates  $B_1, \dots, B_N$  of  $N$  clients, (ii) the loss ratio  $p$  and (iii) the available backbone bandwidth  $B$ , find the optimal streaming strategy, (i.e., the average rates  $\vec{R}$  of layers, respectively versions), such that the average distortion  $D^*$  :

$$D^* = \min_{\vec{R}} \frac{1}{N} \sum_{i=1}^N D_N(\vec{R}), \quad (1)$$

under the constraints that  $\sum_{i=1}^N R_i \leq B$ , and that  $R_i \leq B_i$ .

## 3 Performance in a lossy scenario

This section proposes an analytical model for the performance of the streaming system, in the presence of packet losses on the path between the origin server and the proxy.

Assume that the average distortion  $\tilde{D}$  in the presence of losses can be written as the weighted sum of the source rate distortion  $D$ , and the average distortion due to losses,  $\delta$ , i.e.,  $\tilde{D} = D(1 - \epsilon) + \epsilon\delta$ , where  $\epsilon$  represents the loss probability. This model is quite commonly accepted, and has been shown to provide a reasonable approximation of the distortion in a lossy scenario [3]. The loss degradation  $\delta$  is driven by the size of the area damaged by one loss (it corresponds to one frame in our scenario, where each packet loss induces the loss of a complete frame), and a temporal loss propagation factor  $\eta$ , which depends on the content of the video sequence.

Let  $D_n$  denote the source distortion of the stream encoded at rate  $R_n$ . In the version case, it simply corresponds to the  $n^{th}$  stream, and in the layer case, this stream is the sum of the  $n$  lowest layers. The distortion  $\tilde{D}_1$  experienced by the clients with the lowest access bandwidth, and subscribed to the stream of rate  $R_1$  (i.e., the base layer in the scalable encoding), is thus simply written as :

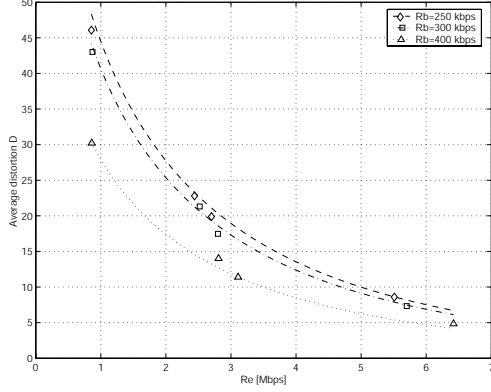
$$\tilde{D}_1 = D_1 (1 - \epsilon_1) + \eta_0 D_0 \epsilon_1, \quad (2)$$

where  $D_0$  represents the average distortion when a complete frame is missing, and  $\eta_0$  is the temporal error propagation factor that can be estimated off-line or for the set of sequences. Under the assumption of a independent packet loss process of probability  $p$ , the frame loss probability  $\epsilon_n$  depends on the encoding rate  $R_n$  (respectively  $r_n$  in the layered encoding) and can be written as  $\epsilon_n = 1 - (1 - p)^{\frac{R_n}{K \cdot S}}$  (respectively  $\epsilon_n = 1 - (1 - p)^{\frac{r_n}{K \cdot S}}$ ), where  $S$  is the average packet size, and  $K$  is a constant given by the frame rate.

In the version streaming scenario, the proxy replaces the lost frames by the corresponding frame taken from a stream at a lower rate. It chooses the closest inferior resolution where the encoded frame is available. If all the versions of a given frame are missing, it cannot be replaced, and the client freezes the previous frame. In general, the distortion  $\tilde{D}_n$ , experienced by the clients subscribed to the stream of rate  $R_n$ , is written as :

$$\begin{aligned} \tilde{D}_n = & D_n (1 - \epsilon_n) + \eta_0 D_0 \prod_{i=1}^n \epsilon_i \\ & + \sum_{j=1}^{n-1} \eta_j D_j (1 - \epsilon_j) \left( \prod_{i=j+1}^n \epsilon_i \right). \end{aligned} \quad (3)$$

In the layered encoding scenario, the data from a given layer can only be decoded if all the lower layers are also correctly received. In case of loss, the proxy does not replace the lost information, and does not even transmit the upper layers since they are useless for the receivers. In the general case, the distortion  $\tilde{D}_n$  as seen by the clients that subscribed to the first  $n$  layers (i.e., a stream of rate  $R_n$ ) is



**Figure 2. Source rate-distortion characteristics, for the CIF Foreman sequence.**

given by :

$$\begin{aligned} \tilde{D}_n &= D_n \left( \prod_{i=1}^n (1 - \epsilon_i) \right) + \eta_0 D_0 \prod_{i=1}^n \epsilon_i \\ &+ \sum_{j=1}^{n-1} \eta_j D_j \left( \prod_{i=1}^j (1 - \epsilon_i) \right) \left( \prod_{i=j+1}^n \epsilon_i \right). \end{aligned} \quad (4)$$

## 4 Optimized streaming strategies

### 4.1 Example: MPEG-4 video

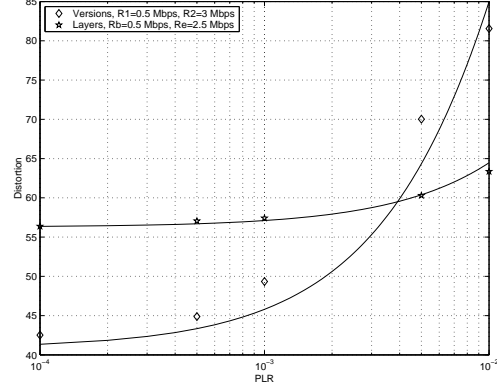
In order to compare version and layer streaming strategies, we now consider the particular case of MPEG-4 FGS video streaming to two clients. We use the Microsoft implementation of the MPEG-4 FGS video coder. Similarly to most coding algorithms, the average source distortion  $D_n$  of a single stream encoded with a rate  $R_n$  can be written as [3]:

$$D_n = \chi R_n^\xi, \quad (5)$$

where  $\chi$  and  $\xi$  are directly related to the encoding scheme and the video sequence content. The average distortion for a layered stream encoded with MPEG-4 FGS can be modelled as :

$$D_n = \chi(r_B)^\xi 2^{\gamma_1 r_E + \gamma_2 \sqrt{r_E}}, \quad (6)$$

where  $(\chi, \xi)$  and  $(\gamma_1, \gamma_2)$  are driven by the scene content. The model proposed in (6) is based on the study proposed in [1], where the influence of the base layer encoding has been factored in through the multiplicative term  $\chi r_B^\xi$ . Figure 2 shows the validity of the source rate-distortion characteristics for MPEG-4 video scheme for medium base layer rates. It is worth noting here that FGS is in general penalized in terms of compression performance compared to a single stream encoding. This has been reported by numerous studies, and also verified for other types of scalable coders [5]. In case of loss, the decoder simply freezes the last correctly received frame, as it is the case in most current decoders. The Figure 3 validates the distortion model



**Figure 3. Total distortion for lossy streaming to two clients, and frame substitution at the proxy (CIF Foreman sequence).**

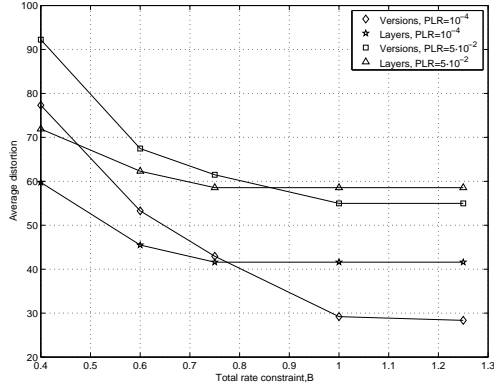
proposed in the previous section in the lossy streaming scenario. It can be seen that switching among streams generates quite acceptable degradations, as long as the loss ratio stays small. Also, it can be seen that, even if the version policy performs better at low loss ratio, the increase of the distortion with the loss probability is much faster than in the layered scenario.

### 4.2 Experimental Results

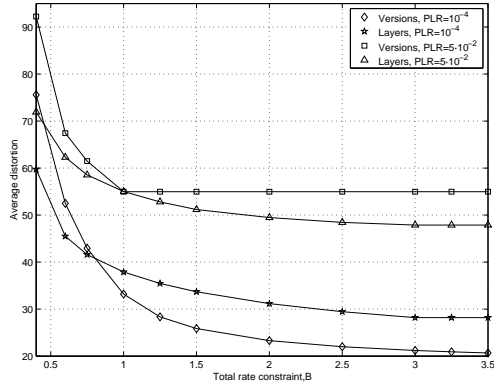
B	Versions		Layers	
	$R_1^{opt}$	$R_2^{opt}$	$R_1^{opt}$	$R_2^{opt}$
0.4	0.2	0.2	0.4	0
0.6	0.29	0.31	0.5	0.1
0.75	0.36	0.39	0.5	0.25
1	0.5	0.5	0.5	0.25
1.25	0.5	0.75	0.5	0.25

**Table 1. Optimal streaming rates [Mbps] in function of  $B$ , for  $p = 10^{-4}$  and  $(B_1, B_2) = (500kbps, 750kbps)$ .**

The optimal streaming strategies for version and layer policies, as given in solving the optimization problem of Eq. (1), are now compared in different scenarios. Table 4.2 first compares the optimal encoding rates as a function of the constraint on the total rate. It can be seen that the version strategy generally tends to evenly distribute the total rate between both clients, while the layer policy generally allocates the maximal possible bandwidth to the base layer. Figure 4 represents the minimal average distortion, as a function of the total rate constraint. When the rate constraint is restrictive, layer streaming performs better, since it clearly presents less redundancy. However, when the available rate is larger, version streaming performs better thanks to a better error resilience. Interestingly, when the error rate is very high (e.g.,  $5 \cdot 10^{-2}$ ), the version policy is always worse than the layer strategy, even for large available bandwidths (ex-



(a)  $(B_1, B_2) = (500kbps, 750kbps)$



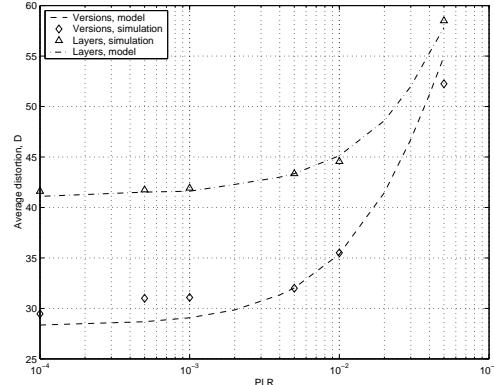
(b)  $(B_1, B_2) = (500kbps, 3Mbps)$

**Figure 4. Minimal distortion, as a function of the total rate constraint.**

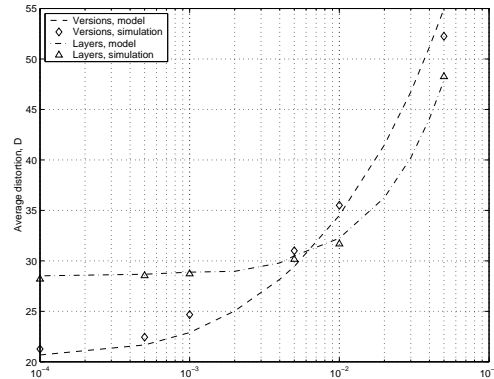
cept when we have the saturation effect in layer case that comes from bandwidth constraint, see Figure 4(a). In this situation, the version streaming policy tries to strictly limit the rate of both versions in order to lower the probability of losing frames. Figure 5 finally compares the performance of the optimal streaming strategies as a function of the loss probability. The version policy performs better when the rate constraint is relaxed, and the access rates are similar. When these access rates are very different, the layer strategy performs better for high loss ratios, as it has been previously observed.

## 5 Conclusions

This paper presented an optimal streaming strategy for both versions- and layers-coded video for transmission over lossy, possibly rate-constrained networks. It is shown that the optimal version policy shall be preferred over the optimal layer streaming strategy, especially at low-to-medium PLRs and large available backbone rates. In future work, we will analyze the impact of adding optimal FEC to our framework. Conclusions drawn from this work should intuitively hold.



(a)  $(B, B_1, B_2) = (1.25Mbps, 500kbps, 750kbps)$



(b)  $(B, B_1, B_2) = (3.5Mbps, 500kbps, 3Mbps)$

**Figure 5. Minimal distortion, as a function of the packet loss probability.**

## References

- [1] Dai M. and Loguinov D. Analysis of rate-distortion functions and congestion control in scalable internet video streaming. In *Proceedings of NOSSDAV*, pages 60–69, 2003.
- [2] de Cuetos P., Saporilla D. and Ross K.W. Adaptive streaming of stored video in a TCP-Friendly Context: Multiple Versions or Multiple Layers ? In *Proceedings of the International Packet Video Workshop*, 2001.
- [3] Frossard P. and Verscheure O. Joint Source/FEC Rate Selection for Quality-Optimal MPEG-2 Video Delivery. *IEEE Transactions on Image Processing*, 10(12):1815–1825, December 2001.
- [4] Kim T. and Ammar M.H. A Comparison of Layering and Stream Replication Video Multicast Schemes. In *Proceedings of NOSSDAV*, June 2001.
- [5] Kimura J.-i., Tobagi F.A., Pulido J.-M. and Emstad P.J. Perceived quality and bandwidth characterization of layered MPEG-2 video encoding. In *Proceedings of the SPIE International Symposium on Voice, Video and Data Communications*, Boston, MA, 1999.
- [6] Yang, Y.R. Min Sik Kim Lam, S.S. Optimal partitioning of multicast receivers. In *Proceedings of ICNP*, pages 129–140, November 2000.