PLAYBACK DELAY OPTIMIZATION IN SCALABLE VIDEO STREAMING

Jean-Paul Wagner, Pascal Frossard

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Institute - LTS4
CH-1015, Lausanne

ABSTRACT

This paper addresses the problem of optimizing the playback delay experienced by a population of heterogeneous clients, in video streaming applications. We consider a typical broadcast scenario, where clients subscribe to different portions of a scalable video stream, depending on their capabilities. Clients share a common channel, whose limited rate directly drives the playback delays imposed to the different groups of receivers. We derive an optimization problem, that targets a fair distribution of the playback delays among heterogeneous clients. A server-based scheduling strategy is then proposed, that takes into account the properties of the targeted clients, the channel status, and the structure of the media encoding. It is shown to offer significantly reduced playback delays per client population, as compared to traditional scheduling strategies. In the same time, PSNR performance is not affected, which altogether leads to an overall improvement of the quality of service.

1. INTRODUCTION

Internet video streaming applications usually make use of client buffering capabilities to smooth the discrepancies between the video source rate, and the available channel bandwidth. Buffering then allows for a smooth playback of the stream, but it generally induces a playback delay at the client, and thus impacts the general quality of service.

The particular problem we consider in this paper consists in a broadcast scenario where scalable media is streamed to a variety of heterogeneous clients, such as smart phones, notebooks or workstations. Due to their different capabilities, these clients subscribe to different resolutions of the media stream. They however share a common broadcast channel, whose limited rate directly affects the resolution of the stream that can be sent, and the playback delay induced by buffering at the client. The order in which data from the different layers are sent by the server directly influences the distribution of the playback delay among the different receivers groups. The server may decide to first send the lower resolution data, or base layer, and thus to favor the least powerful clients, whose playback delay is then minimal. Such a policy however highly penalizes the other groups of clients, that receive an important share of base layer before the enhancement layers, resulting in an increased playback delay.

In this paper, we propose a server-based scheduling strategy that targets a fair distribution of the playback delay among the different groups of clients. It takes into account the network status, the client capabilities, and the video stream characteristics to optimize an average quality of service for all the subscribers. To the best of our knowledge, this work is a first effort to address the playback delay optimization problem for heterogeneous clients.

The paper is organized as follows: we provide an overview of the used system in Section 2. In Section 3 we formalize the considered problem and discuss its implications. Section 4 shows our simulation results. Finally we conclude with Section 5.

2. SCALABLE VIDEO STREAMING

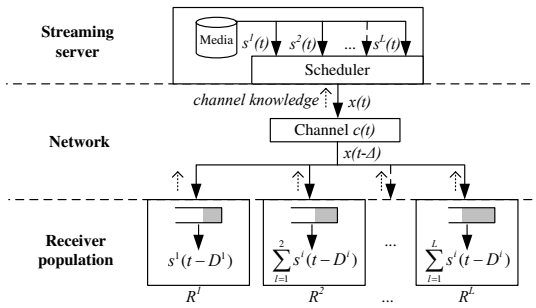

Fig. 1. General overview of the system under consideration.

We consider L-layered hierarchically coded bitstreams that are stored on a streaming server (see Figure 1). In such coding scenarios, all inferior layers from 1 up to l must be present at the decoder in order to decode layer l. Depending on the encoding choice, adding a layer may icrease the PSNR of the decoded video, the framerate, or the framesize.

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

To appear in *Proc. IEEE ICME 2005, Amsterdam, The Netherlands, July 2005.*

Each layer is completely determined by its source trace, or playout trace $s^l(t), 1 \leq l \leq L$, indicating how many bits the layer consumes at all time instants t. The channel connecting the server to the receivers is defined by its bitrate $c(t)$, indicating how many bits the channel is able to transmit at any time t, and a potential network latency Δ. Generally, the server's channel knowledge is extracted from client or network feedback. In this paper we will assume perfect channel knowledge at the server, as offered by guaranteed services for example. For other types of service, it leads to an upper bound on performance. L sets of receivers connect simultaneously to the media stream, where each set $R^l, 1 \leq l \leq L$ groups clients that subscribe to all layers up to l of the media stream.

In such scenarios, the most important logical part of the streaming server is the scheduler: given the source trace and the channel knowledge, it decides when to send data, in order to meet criteria such as desired distortion or delay [1] [2], or maximum utilization of the available channel bitrate. The scheduler outputs a stream of rate $x(t) \leq c(t), \forall t$, the *sending rate*, indicating how many bits are sent on the channel at a given time.

After the first bit of the stream is sent by the server, a client in population R^l waits for a time D^l, during which it buffers the data it receives, to ensure that its receiver buffer will never underflow, i.e., the playback will not be disrupted. We call $\mathcal{D} = \{D^l\}_{l=1}^L$ the set of playback delays at the clients.

We will use capital letters (S, C, X) for the cumulative rate functions, e.g., $C(t) = \int_0^t c(u)du$ is the number of bits the channel can transmit up to time t. Note that the cumulative rate functions are all non-decreasing in t. Using this notation [3][4], Figure 2 illustrates de concept of playback delay. If the client starts playback at the reception of the first bit, a buffer underflow occurs at time t_c. Starting playback at the client after D makes sure that the buffer underflow does not occur. We say that a trace $s(t)$ is schedulable over a given channel $c(t)$, with a playback delay D, if the following condition holds for all t:

$$S(t-D) \quad \leq \quad C(t-\Delta) \tag{1}$$

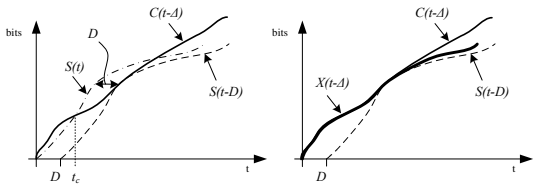

Fig. 2. *Left:* Playback delay and buffer underflow prevention. *Right:* Schedulable play-out trace and a corresponding sending rate trace.

If this condition is met, this implies that the server can find a scheduling such that each of the following, necessary conditions are satisfied for all t:

$$S(t-D) \quad \leq \quad X(t-\Delta) \tag{2}$$
$$X(t) \quad \leq \quad C(t) \tag{3}$$
$$x(t) \quad \leq \quad c(t). \tag{4}$$

We will however not further discuss particular scheduling solutions here. In the remainder, we choose to use the scheduling policy that sends data at the latest possible opportunity given their decoding deadline, so that the overall buffer occupancy is kept small. We also set $\Delta = 0$, without loss of generality.

3. PLAYBACK DELAY OPTIMIZATION

3.1. Problem Formulation

Consider a channel given by its cumulative rate trace $C(t)$, and a set of L hierarchically coded layers given by their cumulative source rate traces $\{S^l\}_{l=1}^L$. The channel connects a streaming server to L sets of receivers $\{R^l\}_{l=1}^L$, that simultaneously subscribe to layers up to l. Our aim is to find a set of playback delays $\mathcal{D} = \{D^l\}_{l=1}^L$, $D^1 \leq D^2 \leq \ldots \leq D^L \leq D_{max}$, that minimizes a global metric $\varphi(\cdot)$ over the set of possible playback delay sets:

$$\mathcal{D}_f \quad = \quad \arg\min_{\mathcal{D}} \left(\varphi(D^0, ..., D^L)\right) \tag{5}$$

such that for any $l \leq L$, $S^l_{\mathcal{D}}(t) \leq C(t)$, where $S^l_{\mathcal{D}}(t) = \sum_{i=1}^l S^i\left(t - D^i\right)$. This is, from (1), a sufficient condition for the trace $S^l_{\mathcal{D}}(t)$ to be schedulable over the channel $c(t)$. Let D^l_{min} denote the smallest possible playback delay for layers up to l. In order to have a fair distribution of the penalty on the playback delay, we choose to minimize the standard deviation of the relative penalties induced by a set of playback delays $\mathcal{D}$:

$$\varphi(D^0, ..., D^L) = \sigma\left((D^0 - D^0_{min}), ..., (D^L - D^L_{min})\right). \tag{6}$$

3.2. Playback Delay Analysis

Suppose we have two increasing non-zero functions $F(t)$ and $G(t)$ such that $\lim_{t\to\infty} F(t) \geq \lim_{t\to\infty} G(t)$. We define the (maximum) horizontal distance between $F(t)$ and $G(t)$ as follows:

$$h(G, F) \quad = \quad \sup_t \left(F^{-1}\left(G(t)\right) - t\right), \tag{7}$$

where $F^{-1}(t) = \min\{t : F(t) \geq x\}$ is a pseudo-inverse of $F(t)$. The following relations hold:

$$h(G, F) = 0 \quad \Leftrightarrow \quad F(t) \geq G(t), \forall t \text{ and} \tag{8}$$
$$\exists \tau \text{ s.t. } F(\tau) = G(\tau) \tag{9}$$
$$h(G, F) < 0 \quad \Leftrightarrow \quad F(t) > G(t), \forall t \tag{10}$$
$$h(G, F) > 0 \quad \Leftrightarrow \quad \exists \tau \text{ s.t. } F(\tau) < G(\tau). \tag{11}$$

Two useful properties of $h(\cdot)$ will be used in the optimization :

1. If $h(G,F) > 0$ and $G'(t) = G(t - h(G,F))$, then $h(G',F) = 0$. In other words, $h(G,F)$ is the minimum shift we need to apply on $G(t)$, so that $F(t) \geq G'(t)$, $\forall t$.

2. Let $F(t)$, $G(t)$ and $G'(t)$ be non-decreasing functions such that $G'(t) > G(t)$, $\forall t$. Then: $h(G',F) > h(G,F)$. Indeed by the definition of $h(\cdot)$ and $F^{-1}(\cdot)$, and because $F(t)$ is non-decreasing, the result follows immediately, as $F^{-1}(G'(t)) > F^{-1}(G(t))$, $\forall t$. Similarly, if $G'(t) < G(t)$, $\forall t$ then $h(G',F) < h(G,F)$.

Let $\vec{\delta}$ denote any set of L decreasingly ordered positive values: $\vec{\delta} = \{\delta_1, \delta_2, \ldots, \delta_L\}$, $\delta_1 \geq \delta_2 \geq \ldots \geq \delta_L \geq 0$. We will use the following notation $\forall l$, $1 \geq l \geq L$: $G^l_{\vec{\delta}}(t) = \sum_{i=1}^{l} G^i(t + \delta_i)$ and $G^l_0(t) = \sum_{i=1}^{l} G^i(t)$.

Lemma 3.1 *Consider a set of L non-decreasing functions $\{G^l(t)\}_{l=1}^L$ and a non-decreasing function $F(t)$, all defined on the temporal axis. We have, $\forall l$, $1 \geq l \geq L$ and $\forall \vec{\delta}$:*

$$D^l_0 = h\left(G^l_0, F\right) \quad < \quad h\left(G^l_{\vec{\delta}}, F\right) = D^l_{\vec{\delta}} \qquad (12)$$

Proof As the functions $\{G^l(t)\}_{l=1}^L$ are non-decreasing, we have, $\forall l$, $1 \geq l \geq L$ and $\forall \delta_l > 0$: $G^l(t) < G^l(t + \delta_l)$. Thus, $\forall l$, $1 \geq l \geq L$ and $\forall \vec{\delta}$:

$$G^l_0(t) \quad < \quad G^l_{\vec{\delta}}(t), \forall t \qquad (13)$$

From above, it follows that $D^l_0 < D^l_{\vec{\delta}}$. ∎

3.3. Discussion

Applying this property to cumulative source rate traces, we derive a lower bound on the playback delay for the clients in set R^l. If $S^l_0(t) > C(t)$, for some t, the smallest playback delay for layer l, is given by:

$$D^l_0 \quad = \quad h\left(S^l_0, C\right), \qquad (14)$$

where $S^l_0(t) = \sum_{i=1}^{l} S^i(t)$ is the sum of all the layers without relative shifts. Since layers are hierarchically ordered, we know, by application of Lemma 3.1, that D^l_0 is the lower bound on all possible playback delays for layer l, thus $D^l_{min} = D^l_0$. Furthermore, as all the rate traces are positive valued functions, $S^{l+1}_0(t) \geq S^l_0(t)$, $\forall t$, so from (3.2) we have $D^{l+1}_{min} \geq D^l_{min}$.

It is important to note that, by achieving the minimum playback delay for a given layer l, we do not necessarily achieve the minimum playback delay for any other layer. This can be illustrated using a simple 2-layer example, depicted in Figure 3. In Figure 3-top, we set $D^1 = D^1_{min}$, without considering higher layers. In that case playout of layer 1 can begin after $D^1_{min} = 2$ frames. If we consider layer 2 (Figure 3-middle) without taking into account lower layers individually, playout of layers 1 and 2 can begin after $D^2_{min} = 127$ frames. Note the induced playback delay penalty of 125 frames for clients in set R^1. Figure 3-bottom: in the case where D_1 is fixed to D^1_{min} the bitrate available for layer 2 is given by $C^2(t) = C(t) - X^1(t)$, $X^1(t)$ being the received rate, as computed by the scheduler. Under these conditions, the playback delay for layer 2 grows to 219 frames, inducing a relative penalty of 92 frames. We are finally facing a typical tradeoff situation: as we increase the relative playback delay penalty for lower layers, we leave more available channel bits that can be used to decrease the playback delay penalty for higher layers.

3.4. Optimization Algorithm

Finding $\mathcal{D}_f = \{D^l_f\}_{l=1}^L$ is a combinatorial optimization problem over the integer domain of delays, and solving it generally implies a full search algorithm. Based on the example introduced above we can however make the following observations about the structure of the solution space. First, from the greedy rate allocation scheme illustrated in Figure 3, we see that we can easily compute the largest (greedy) playback delay D^L_g for the highest layer L. Indeed, fixing each $D^l_g, l < L$ to the shortest possible delay, given the available bitrate $C^l(t)$ which results from the same procedure on the previous layer, all the spare bitrate for layer L

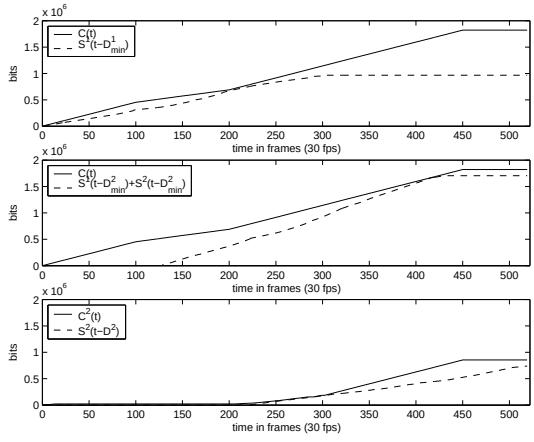

Fig. 3. Scheduling 300 frames of Foreman (QCIF). The 2 layers represent the MPEG-4 FGS base layer and the first bitplane of the enhancement layer. The channel is available for 450 time units, providing a mean rate of 128kbps, which drops to 64kbps between times 100 and 200.

will be available at the latest possible instant in time. Then, we also have seen that we can easily compute D^L_{min}. We can thus drastically limit the range of values in which D^L can evolve. Starting from there, we can loop through the possible values of D^L in the range $[D^L_{min}, D^L_g]$. Fixing a value D^L, we compute the bitrate available for layers 1 to $L-1$ as: $C^{L-1}(t) = C(t) - X^L_g(t)$. Given this channel bitrate, we then compute the range of possible delays for layer $L-1$, $[D^{L-1}_{min}, D^L]$, fix a value D^{L-1} and iterate down through the lower layers. While searching, we further use the fact that, for a fixed D^l we have $D^{l+1} \geq D^l$, thus limiting the search space to only feasible solutions. We can even reduce the number of iterations by using a simple threshold on $\varphi(\cdot)$ to get an approximation of the minimum.

4. RESULTS

	D^2_{min}	D_f
D^1	212	54
D^2	212	259

	D^2_{min}	D_f
D^1	241	59
D^2	241	297

Table 1. Fair playback delays D_f in frame units, compared to D^2_{min}. *Left:* 2 layers of Foreman over a 100kbps channel. *Right:* same channel, 2 layers of the composite sequence.

The video traces we used in our simulations are 300 frames of Foreman, and a composite sequence made up of 300 frames of Foreman, followed by 300 frames of Coastguard and 300 frames of News, all QCIF at 30 frames per second. We used the MoMuSys MPEG-4 FGS [5] reference codec to encode the sequences into a base layer and an enhancement layer. The *GOP* size is 150 frames, and it only contains P-frames. The enhancement layer has been cut along the bitplane boundaries to construct further layers.

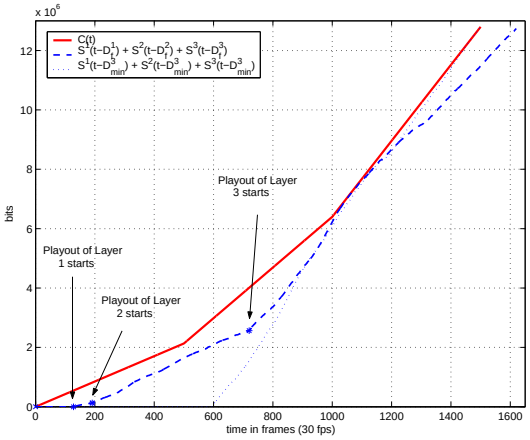

Fig. 4. The channel can support 3 layers of the encoded stream. The dashed curve shows the aggregate playout curve of the 3 layers with playback delays $\mathcal{D}_f$. The aggregate playout curve of the 3 layers using playback delay D^3_{min} is shown for reference (dotted line).

Table 1 shows results for both sequences sent over a channel of mean rate 100kbps. The channel can transmit 2 layers in both cases. The gain in playback delay for receivers of set R^1 is of the order of seconds when using our fair distribution. Figure 4 shows the results of another simulation run: we consider sending the composite sequence over a piecewise CBR channel which provides a mean rate of 128kbps at the beginning, then improves to 256kbps and finally to 384kbps. Using a fair playback delay distribution given by $\mathcal{D}_f$, playout can begin after a playback delay of 128 frames at receivers of set R^1. Similarly the playback delays for layer 2 and 3 are of 191 and 720 frames respectively. Note the gain in delay for clients in sets R^1 and R^2, compared to a playback delay of 594 frames if D^3_{min} is used for all clients (dotted line). The relative playback delay penalty per client set, compared to their respective D^l_{min} value, is of 126 frames each.

5. CONCLUSIONS

In this paper, we have outlined and formalized the problem of playback delay distribution in a scalable streaming scenario, where a streaming server broadcasts to a heterogeneous set of clients. We have proposed a server-based scheduling strategy, that targets a fair distribution of the playback delays. It is shown to bring significant improvements on the playback delays experienced by the clients, since it takes into account the heterogeneities in the client population, the structure of the encoded stream, and the available channel knowledge.

6. REFERENCES

[1] P.A. Chou, Z. Miao, *Rate-distortion optimized streaming of packetized media*, Microsoft Research Technical Report MSR-TR-2001-35, February 2001.

[2] P. Decuetos, K.W. Ross *Unified Framework for Optimal Video Streaming*, Infocom 2004, Hong Kong, February 2004.

[3] T. Stockhammer, H. Jenkac, G. Kuhn, *Streaming Video over Variable Bit-Rate Wireless Channels*, IEEE Transactions on Multimedia, Vol. 6, No.2, April 2004.

[4] P. Thiran, J.-Y. Le Boudec, *Network Calculus*, Springer-Verlag Lecture Notes on Computer Science number 2050.

[5] W. Li, *Overview of Fine Granularity Scalability in MPEG-4 Video Standard*, IEEE Transactions on Circuits and Systems for Video Technology, March 2001.