MEDIA STREAMING WITH CONSERVATIVE DELAY ON VARIABLE RATE CHANNELS

Dan Jurca and Pascal Frossard

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

ABSTRACT

We address the problem of delay-constrained streaming of multimedia packets over dynamic bandwidth channels. Efficient streaming solutions generally rely on the knowledge of the channel bandwidth, in order to select the media packets to be transmitted, according with their sending time. However, the streaming server usually cannot have a perfect knowledge of the channel bandwidth, and important packets may be lost because of over-estimation. We address the rate prediction mismatch by media scheduling with a conservative delay, which provides a safety margin for the packet delivery, even in the presence of unpredicted bandwidth variations. We formulate an optimization problem whose goal is to find the optimal conservative delay to be used in the scheduling process, given the network model and the playback delay imposed by the client. We then propose a simple solution to the scheduling delay estimation, effective in real-time streaming scenarios. Our streaming method proves robust against channel prediction errors, and performs better than other mechanisms based on frame reordering strategies.

1. INTRODUCTION

Under timing constraints imposed by a fixed playback delay, efficient media streaming solutions must adapt the media packets scheduling, to the available channel resources, in order to optimize the quality of service at the client. Channel conditions are highly variable, and adaptive scheduling strategies are necessary to compensate for the mismatch between the rate of the media stream, and the available streaming rate. Among the most popular schemes, a first method models the underlying network in a stochastic framework [2] and, given a tolerated playback delay at the client, attempts to maximize the expected received media quality. The second one considers the network topology and parameters as known in advance and realizes a deterministic scheduling of the packets, in order to maximize the received media quality [1, 3]. Contrary to the former method, which transforms the scheduling decision into a stochastic optimization problem that requires complex algorithms, the latter is simpler and can be employed in real-time applications. It is however vulnerable to channel prediction errors.

Even the best rate estimation algorithms are not able to follow the rate variations of the channel, and often work on a coarser timescale [4]. Since channel prediction errors are inevitable and can lead to late arrivals of important media packets, the streaming server has to design robust scheduling strategies against estimation mismatches. Previous works [6] assess the efficiency of a scheduling model in which the frames in a bitstream are rearranged such that the most important ones are sent before the less important ones. While the method increases the robustness to network delay fluctuations, it is also more demanding in terms of codec buffer sizes and computation.

In this work, we rather enforce a FIFO scheduling mechanism because of its simplicity and efficient use of buffering resources. However, we propose to increase its robustness to channel estimation errors by scheduling packets with a conservative virtual playback delay, smaller than the playback delay imposed by the client. The difference between the scheduling delay and the playback delay after which the client starts playing the video, can transparently absorb the effects of the erroneously predicted end-to-end rate variations on packet arrival times. A very conservative scheduling delay tends to limit the selection of transmitted media data to only a few packets, which penalizes the quality at the receiver. Alternatively, a scheduling delay that is too close to the effective playback delay may result in late arrival of packets, which also penalizes the quality. We formulate in Section 2 an optimization problem whose goal is to find the optimal conservative delay used in the scheduling process, which maximizes the quality of the received video for a given channel rate model, and a given playback delay at the client. We discuss the complexity of the exact solution for the optimization problem and we present a fast solution in Section 3. Section 4 presents our simulation results and Section 5 concludes this paper.

2. STREAMING WITH CONSERVATIVE DELAY

2.1. System Overview

We consider a single path streaming scenario between a server S and a client C. The media stream can either be

This work has been supported by the Swiss National Science Foundation, under grant PP002-68737.

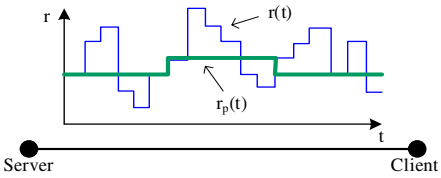

Fig. 1. Network end-to-end Model with Rate Variations $r(t)$ and Estimated Rate $r_p(t)$.

pre-stored at the server (VoD), or can be obtained in real time (real-time streaming). The video content is encoded into one or more layers and fragmented into network packets such that one packet contains information related to one frame and one video layer. Let $P = \{p_1, ..., p_n\}$ be the set of available packets at the server, with n representing the total number of packets. Similarly to [3], each packet p_i is completely characterized by its size s_i, its decoding deadline t_i, its importance ω_i and its list of dependency packets A_i, which are necessary for a correct decoding.

The intermediate network between S and C is modeled as an end-to-end channel characterized by the variable rate $r(t)$. While we consider no link error in our model, packets can still be lost from a media application perspective, due to late arrivals. The server S estimates on a periodic interval, the available channel rate $r_p(t)$, using any estimation mechanism Γ (see Figure 1). Based on that estimation, the streaming application employs a generic scheduling algorithm Ψ that decides the subset of packets $\pi \subseteq P$ that are sent in a FIFO order to the client, so that the reconstructed video quality is maximized, given the playback delay Δ imposed by the client. The video quality measure Ω, can be computed at the client as $\Omega = \Omega_S(\pi) - \Omega_L(\pi)$, where $\Omega_S(\pi) = \sum_i \omega_i, \forall p_i \in \pi$ represents the quality of the video packets selected for transmission, and $\Omega_L(\pi) = \sum_i (\omega_i \cdot P_i)$ represents the video quality degradation due to packets that cannot be decoded because of late arrivals at the client. P_i represents the probability that packet p_i arrives past its decoding deadline at the client. These late arrivals are caused by channel bandwidth variations, and inaccuracy in the rate estimation used by the server. Indeed, the estimation of the available rate in the future time instants is generally not perfect, and often not able to exactly follow the frequent bandwidth variations.

We propose to modify the scheduling strategy, in order to be robust to over-estimations of the channel rate. We define a virtual playback delay, or scheduling delay δ, which is used by the server to compute the subset of packets to be sent. As δ is smaller than the actual playback delay Δ, the server will select a reduced number of packets to be sent (Ω_S decreases), but the selected packets have a lower probability to be lost (Ω_L increases). In other words, π now contains only packets that can reach the client before their decoding deadline ($t_i + \delta$) with a streaming rate r_p, and

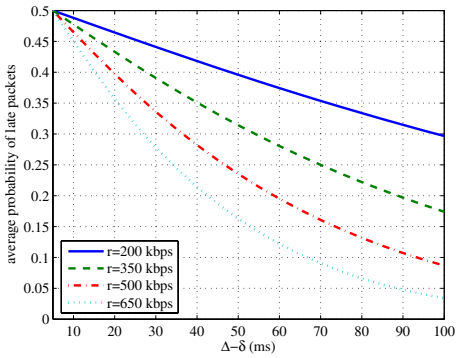

Fig. 2. Average Probability of Late Packets ($\Delta = 300ms$).

each packet p_i is scheduled and transmitted only once. The choice of the virtual playback delay becomes obviously a trade-off between source quality, and robustness to rate variations, and its optimization is proposed in the next section.

2.2. Optimization Problem

The virtual playback delay δ used by the scheduler represents a compromise between a conservative selection of packets, and their probability of late arrivals. Given the video sequence, the quality metric Ω, the scheduling strategy Ψ, the rate estimation algorithm Γ, and the playback delay Δ, the optimization problem translates into finding:

$$\delta^* = \arg \max_{\forall \delta \leq \Delta} \Omega(\delta) \quad (1)$$

In general, this optimization problem does not provide any simple solution. Even for fixed Ψ, Γ and Δ, the scheduling policy π is not constant with the choice of δ, hence finding the optimal solution for the problem has combinatorial complexity. However, for small values of Δ (as in practical real-time streaming scenarios), δ^* can be accurately approximated in real-time. In the next section we present our approach to finding an appropriate solution, based on heuristics from real-time video streaming.

3. FINDING THE CONSERVATIVE DELAY

3.1. General Solution

On one hand, the quality measure $\Omega_L(\pi)$ depends only on the difference $\Delta - \delta$, for a given transmission policy π and the channel model. Very conservative values for δ will insure a big difference $\Delta - \delta$, hence more margin in dealing with rate prediction errors, and consequently a smaller value for Ω_L (see Figure 2).

On the other hand, the quality measure $\Omega_S(\pi)$ depends only on the scheduled packets according to the predicted rate $r_p(t)$ and δ. Interestingly, our experiments show that, for a given channel model, Ω_S does not vary much with δ, as long as δ is large enough to accommodate the transmission of the largest video packet of the sequence.

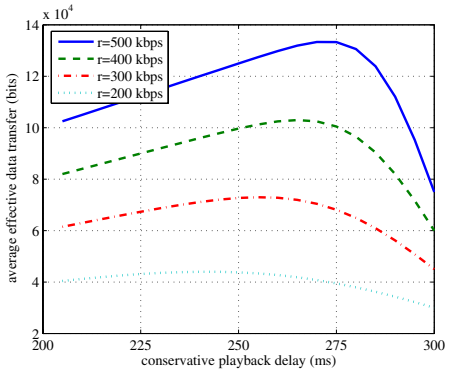

Fig. 3. Effective Average Data Transfer ($\Delta = 300ms$).

Let $R^i(\Delta)$ be the cumulative rate of the channel up to time $t_i + \Delta$: $R^i(\Delta) = \int_0^{t_i+\Delta} r dt$, and $R_p^i(\delta)$ be the cumulative estimated rate up to time $t_i + \delta$: $R_p^i(\delta) = \int_0^{t_i+\delta} r_p dt$. For given δ and Δ, we define on the time interval $[0, t_i + \Delta]$, the effective data transfer $\mathcal{C}_\Delta^\delta(i)$, as the amount of data scheduled according to r_p before $t_i + \delta$, and received before $t_i + \Delta$ according to r: $\mathcal{C}_\Delta^\delta(i) = R_p^i(\delta) \cdot Pr\{R_p^i(\delta) \leq R^i(\Delta)\}$. An illustration of the effective data rate transfer is give in Figure 3.

Given this measure, we transform the original optimization problem, into a new one that chooses δ in order to maximize $\mathcal{C}$, defined as:

$$\delta^* = \arg \max_{0 \leq \delta \leq \Delta} \mathcal{C}_\Delta^\delta(i). \quad (2)$$

$\mathcal{C}_\Delta^\delta(i)$ is invariant in time, as long as the channel model does not change, hence it can be computed at any t_i. The previous optimization problem translates into maximizing the chances of every packet p_i, scheduled for transmission at time t, to reach its destination by time $t + \Delta$. Unlike the original optimization problem of Eq. (1), Eq. (2) depends only on the channel model, hence it is easy to solve, once this model is known. It can be noted that both optimization problems are equivalent in the case of a smooth video model (the video packets have the same size and importance, and there are no dependency among them). In Section 4, we show that even in realistic video streaming scenarios the solution obtained for this problem is a very good approximation of the optimal solution.

3.2. Example Channel Model

We now develop all necessary relations for a typical channel modeled as a discrete-time system, with a sampling interval of T_s seconds. The network can communicate a maximum of $r_i T_s$ bits of data in the time interval $[iT_s, (i+1)T_s]$, where r_i is the available bandwidth of the channel in the i^{th} time interval. The channel rate r_i is given as a Gaussian autoregressive process of the form $r_i = \mu + (1-\alpha) \sum_{j=0}^{\infty} \alpha^j n_{i-j}$, $j \in Z$, $n_k = 0, \forall k < 0$. Each n_j is an independent zero mean Gaussian random variable with variance σ^2, α is a modelling parameter, and μ denotes the average available bandwidth. The validity of that model for internet traffic traces on time scales of milliseconds up to a few seconds has been verified in [5].

A simple auto-regressive prediction model is used for bandwidth estimation at the server, where the available rate of the network in the next time interval, $k+1$, is given by: $r_{k+1} = \gamma \frac{\sum_{j=1}^{k-1} r_j}{k-1} + (1-\gamma) r_k$, where γ is the prediction coefficient. The estimation is run periodically, on time windows of size T_p. While instantaneous rate variations of the channel can happen on very small time scales (of tens to hundreds of milliseconds), the fastest estimation mechanisms provide accurate results on time intervals of the size of a few round-trip times (e.g., one second or more), and prediction inaccuracies cannot be avoided.

Assuming that $t_i + \Delta = k \cdot T_s \leq T_p$, with k an integer[1], we can compute:

$$R^i(\Delta) = k \cdot \mu + \sum_{j=0}^{k} (1-\gamma) \cdot \gamma^{j-1} \cdot \sum_{l=1}^{k-j} n_l$$

Finally, $\mathcal{S}_i$ denotes the cumulative size of the transmitted packets up to packet p_i: $\mathcal{S}_i = \sum_{j=1}^{i} s_j$, $\forall p_j \in \pi$. The probability that a packet arrives too late at the receiver, P_i, can be computed as: $P_i = Pr\{\mathcal{S}_i > R^i / \mathcal{S}_i \leq R_p^i\}$. Since R^i is a normal random variable and R_p^i is a known constant, given any δ and Δ, the error probabilities P_i can be easily computed with the help of the $erfc$ function.

4. SIMULATIONS

We discuss the performance of the streaming application with conservative delay and we compare the results obtained by our heuristic solution for δ with the optimal one, and with other frame reordering techniques. We scalably encode the $foreman\_cif$ sequence (130 frames) using MPEG4-FGS, at 30 frames per second, with a GOP structure of 31 frames ($IPBPBPB...$). By splitting the bitplanes, we encode one BL and 2 ELs of average rates of $260kbps$. In all our experiments we use a simple packet scheduling algorithm on one path, and set the packet weights similar to [3]. For the channel model and estimation mechanism we set the required parameters to $\alpha = \gamma = 0.8$, $T_s = 20ms$, $T_p = 1s$, and we vary $\sigma^2 \in [100, 250]$, according to the channel average rate. These values insure realistic channel variation on small time scales around the average bandwidth value. Finally, we set $\Delta = 200ms$.

First we compare the results obtained by streaming with the heuristic δ, computed according to Eq. (2), and the optimal δ^*. We use different channel average rates and we

[1]The extension of the computation for the general case, on multiple prediction intervals, and when k is not an integer, is straightforward, and omitted due to lack of space.

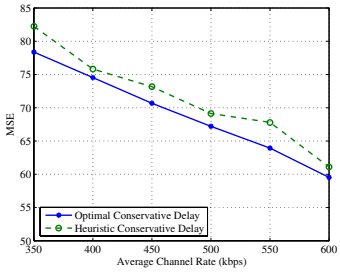

Fig. 4. Quality Evaluation for Scheduling with Heuristic and Optimal δ.

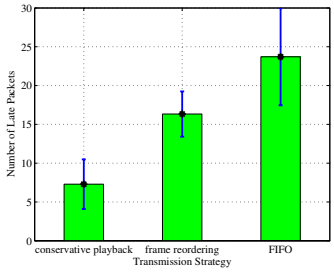

Fig. 5. Late Packets: Conservative δ; Frame Reordering; FIFO Scheduling.

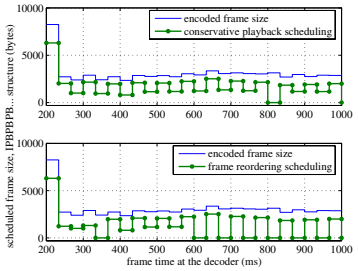

Fig. 6. Example of Conservative δ and Frame Reordering Scheduling.

Table 1. δ^* and δ for Various Average Channel Rates.

Rate ($kbps$)	350	400	450	500	550	600
Optimal δ^* (ms)	163	156	172.5	161	154	155.5
Heuristic δ (ms)	172	170	168	167	166	165
$\frac{\Omega(\delta^*)-\Omega(\delta)}{\Omega(\delta^*)}$ (%)	4.94	1.71	3.53	2.86	6.04	2.63

average over 10 simulations for each case. The results are presented in Figure 4. We observe that for all simulated rates, our results in terms of MSE are very close to the optimal ones. This validates our simplification to the original optimization problem, presented in Section 3. In the same time, Table 1 presents the obtained values for the heuristic and optimal δ for the same channel conditions as above, along with the relative error between the streaming performance. We observe that the values are very close and that δ^* is in general more conservative than δ. An explanation to this phenomenon resides in the fact that the sequence under consideration does not present any scene changes and the packet sizes remain constant in time.

Next, we compare the proposed conservative δ streaming with other frame reordering streaming techniques. We use a simple technique similar to the one presented in [6], which brings forward all I and P frames by two positions in the original bitstream before scheduling. Both techniques are compared in terms of number of late packet arrivals with a simple FIFO scheduling scheme that is unaware of channel rate variations. Simulation results are averaged over 100 channel realizations for an average rate of $500kbps$. Figure 5 presents the number of late packets for each of the 3 schemes with the 95% confidence intervals. We observe that the conservative δ scheme performs the best in terms of average number of late arrivals, due to the fact that the application can transparently use the difference $\Delta - \delta$ to compensate for unpredicted channel rate variations. Finally, Figure 6 presents one scheduling example for the conservative δ and frame reordering techniques. We observe that in the case of frame reordering, the strategy trades off a higher confidence in receiving I and P frames on time, at the expense of less important B frames. Hence, some B frames are lost due to late arrivals. On the contrary, the conservative δ strategy manages to schedule a similar amount of packets, and uses the extra time $\Delta - \delta$ to minimize the impact of rate variations on late arrivals. Hence, less packets are late at the receiving end of the application.

5. CONCLUSIONS

We present a new mechanism to improve the robustness of adaptive media stream scheduling algorithms against network channel variability and estimation inaccuracies. By using a conservative virtual playback delay in the scheduling process we compensate for eventual prediction errors. The difference between the conservative and actual delay imposed by the client transparently absorbs the negative effects of inexact rate estimation (e.g., increased packet delay at the client due to channel variations). We propose a method to determine the value of the conservative delay, as a trade-off between source quality, and robustness to bandwidth variations. The proposed solution is generic and can be employed with any given streaming mechanism. The simplicity of our solution and its effectiveness make it appropriate for any real-time streaming mechanism over best-effort networks.

6. REFERENCES

[1] K. Chebrolu and R. Rao. Bandwidth aggregation for real-time applications in heterogeneous wireless networks. *IEEE Transactions on Mobile Computing*, 2005, accepted for publication.

[2] P. A. Chou and Z. Miao. Rate-distortion optimized streaming of packetized media. *IEEE Transactions on Multimedia*, 2005, accepted for publication.

[3] D. Jurca and P. Frossard. Distortion optimized multipath video streaming. In *Proceedings of Packet Video Workshop*, 2004.

[4] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell. pathChirp: Efficient available bandwidth estimation for network paths. In *Proceedings of Passive and Active Measurement Workshop* , April 2003.

[5] A. Sang and S.-Q. Li. A predictability analysis of network traffic. In *Proceedings of IEEE INFOCOM*, 2000.

[6] S. Wee, W.-T. Tan, J. Apostolopoulos, and M. Etoh. Optimized Video Streaming for Networks with Varying Delay. In *Proceedings of IEEE Conference on Multimedia and Expo (ICME'02)*, July 2002.