ADAPTIVE VIDEO STREAMING WITH LONG TERM FEEDBACKS

Nicolas Tizon, Béatrice Pesquet-Popescu, Marco Cagnazzo

TELECOM ParisTech, CNRS LTCI, Signal and Image Processing Department
46 rue Barrault, 75634 Paris Cedex 13, France
Email: {tizon,pesquet,cagnazzo}@telecom-paristech.fr

ABSTRACT

This paper proposes a video streaming system optimizing resource utilization when the media server only disposes of long term feedbacks from the client. Based on a partial knowledge of the network, we developed a scheduling algorithm that exploits the scalable video coding (SVC) properties to estimate packets importance and that takes into account packet delay dependencies to better anticipate congestion situations. Compared to more conventional streaming systems, experimental results show that our approach allows to better face network condition degradation like bandwidth reduction or packet error rate increase.

Index Terms— Video streaming, scalable coding, hierarchical scheduling, playout deadline, congestion control.

1. INTRODUCTION

Bitrate adaptation is a key issue when considering streaming applications involving throughput limited networks with error prone channels, as wireless networks. Concerning the availability of real time values of the network parameters like resource allocation among users or channel error rate, it is worth noticing that today, in a majority of practical cases, the media server is far away from bottleneck links, thus preventing real time adaptation. Classically, only long term feedbacks like RTCP reports can be used to perform estimations. To our knowledge, [1] is one of the most developed frameworks, that allows to control sent data rate, inferring network and client parameters based on RTCP feedbacks. The main purpose of the described algorithm is to avoid packet congestion in the network and client buffer starvation. In this paper, we take up the idea that RTCP provides relevant information to avoid network congestion and more particularly to infer parameter values of a simple but efficient network model.

In the scope of packetized media streaming over best-effort networks and more precisely channel adaptive video streaming, Chou *et al.* in [2] described a rate-distortion optimized (RaDiO) scheduling algorithm that has been extended in many recent works. Several works proposed the minimization of a rate-distortion cost function in different contexts and applied to scalable coded streams. For example, in [3], the RaDiO scheme is further investigated for video streaming through a shared communication link. In this case, the goal is to minimize a rate-distortion cost over a set of several users. In [4] an unequal error protection (UEP) scheme is described using SVC and assuming that the server is omniscient concerning radio link control (RLC) protocol parameters and frame loss rate. Similarly, in [5], the authors use SVC jointly with UEP schemes based on low-density-parity-check (LDPC) codes. In each case, the distortion model implies intensive computation of mean square errors. In this paper, we will use SVC properties [6] in order to infer a modular scheme to compute video packets importance and their relative priorities.

In [7], the error process of a wireless fading channel is approximated by a first order Markov process. Then, the server uses this model combined with video frame based acknowledgment (ACK/NACK) from the client to compute the expected distortion reduction to be maximized. In the same way in [8], we have proposed an advanced video streaming system based on SVC with Region Of Interest (ROI) to adapt transmission to channel conditions with retransmission mechanisms at radio protocol level. One of the main contributions of this study is to optimize client experienced quality fighting against network congestion caused by the capacity of one user stream to saturate the capacity of the channel. This point of view is quite similar than the one proposed in [9], but our solution takes into account packet delay dependencies and uses a substream based structure in order to anticipate congestion in the network.

The paper is organized as follows. In the next section, the network architecture is detailed and the parameter estimation is described. Then, in section 3 we propose a scheduling algorithm. Experiments results are presented in Section 4 to evaluate the approach and finally we conclude in Section 5.

2. NETWORK MODEL AND PROTOCOLS

The main idea of our algorithm is to maintain the video playout continuity at client side preventing network congestion thanks to well established network protocols and complying with the OSI model. Hence, jointly with the history of sent data, RTCP Receiver Report (RTCP-RR) [10] packets transmitted by the client to the server provide an efficient tool in order to estimate network congestion. As illustrated in Fig. 1, we consider a classical client-server architecture in which a bottleneck link, like a wireless channel, accumulates data at its entry point. We model this entry point as a buffer containing RTP packets and where packets life is governed by a time-out policy. For simplicity, we also consider that the delay for a packet to reach the bottleneck queue from the video server is null. In terms

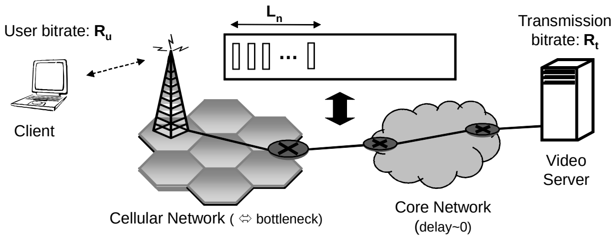
User bitrate: $\mathbf{R_u}$
$\mathbf{L_n}$
Transmission bitrate: $\mathbf{R_t}$
Client
Video Server
Cellular Network (⇔ bottleneck)
Core Network (delay~0)

Fig. 1. Network architecture.

of memory requirements, we assume that the server stores the size of each transmitted RTP packet together with their corresponding timestamp and sequence number. In RTCP-RR, the field “fraction lost” indicates the fraction of RTP data packets lost since the previous RR packet was sent. In our algorithm, to evaluate precisely the effective video bitrate received by the user and denoted R_u in the sequel, we take only account packets that are received before their deadline by the client. Moreover, the server keeps in memory the value of the “extended highest sequence number received” field of the last received RTCP-RR. Then, receiving the next RTCP-RR and using the history of sent packets, the server is able to calculate the bitrate R_u of data arrived at the user decoder.

As illustrated in Fig. 1, let us denote by L_n the total size of packets accumulated in the network and waiting to be transmitted through the network channel. From the estimation of the two parameters R_u and L_n, the latency of the network can be computed as follows:

$$t_{latency} = L_n / R_u \tag{1}$$

This latency is then compared to a threshold value $T_{congest}$ empirically fixed. Giving the result of this test, the scheduling algorithm will operate in one of the two following modes:

- Not congested ($t_{latency} < T_{congest}$): the capacity of the channel is not efficiently exploited and the algorithm takes the decision to increase the transmission bitrate $R_t = R_u(1+\alpha)$.
- Congested ($t_{latency} \geq T_{congest}$): the maximum capacity is reached and the network is going to accumulate data. The server transmission rate is $R_t = R_u$.

In the not congested case, the α coefficient allows to gradually increase the transmission bitrate. The value of this coefficient depends on the congestion level, measured by the ratio between the latency and the threshold value as follows:

$$\alpha = \alpha_{max}(1 - t_{latency}/T_{congest}), \tag{2}$$

where α_{max} determines the increasing rate of the transmission bitrate before reaching the congested state. When the congestion level is low, the bitrate is significantly increased and when the congestion limit is reached, the increase is slowered in order to avoid saturating the channel with the consequence of undesirable packet loss. In Section 4, based on simulation results, we discuss different strategies to set α_{max} and $T_{congest}$ values. In the next section, exploiting the SVC bistream structure, we introduce a distortion model and then propose a packet scheduling algorithm that relies on previously defined network parameters.

3. DISTORTION AND TRANSMISSION POLICY

3.1. Distortion model

The problem which consists in evaluating the contribution of a video packet to the user experienced quality in the case of non-scalable coding is very complex, as it depends on the underlying temporal dependencies and concealment strategy. In the remaining of this paper, we focus on temporal and SNR combined scalability domains and for simplicity, we assume that the base quality layer packet of each video frame at the maximum available frame rate is received at time by the decoder. This assumption guarantees that each video frame will be decodable by the client. We also define $L = T \times Q$, the number of scalability levels we can obtain combining T temporal levels and Q quality layers.

Then, let us consider the distortion decrease caused on the overall video when the packet of the l^{th} scalable level of the n^{th} decoded video frame is present in relation with the video distortion obtained when the packet is absent. Calculating for each packet the corresponding distortion is computationally intensive. However, in the case of an SVC stream we can consider that scalable layers define equivalent classes, in the set of the well known NAL units [6], in terms of distortion contributions. In others words, we assume that each NAL unit of the same combined temporal and SNR layers equivalently contribute to the final distortion:

$$\forall (l,n) \in \mathbb{N}^2, D^l(n) = \begin{cases} D^l & \text{if the packet is lost} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

In the following, we also consider that each RTP payload is constituted by exactly one NAL unit. Then, let us define f a bijection from $\mathbb{N}^2$ to $\mathbb{N}$ which associates a given (t,q) couple of temporal and SNR scalability indices to a combined scalability level index $l = f(t,q)$. In terms of decoding dependencies, lower values of t and q indicate packets with higher priorities. Comparing packets according to their contribution to the final distortion is straightforward when they belong to the same temporal level or to the same SNR layer: $l_1 = f(t_1,q_1), l_2 = f(t_2,q_2)$ and $t_1 = t_2$ or $q_1 = q_2$.

When the scalability layers of the two domains are different ($t_1 \neq t_2$ and $q_1 \neq q_2$), packet classification is more complicated and usually no longer depends on mean square error measures but also on the application requirements. To free ourself of this difficulty and to provide more flexibility to our algorithm we define S substreams, as illustrated if Fig. 2, that not only correspond to a unique temporal level value t but can contain all (q,t) indexed packets with q fixed and t verifying a given relation like $t_1 < t < t_2$ for example. In Section 4 we will fix $0 < t < t_{max}$ with t_{max} being the highest temporal level. Hence, a packet is uniquely designed by the frame number and the substream index it belongs to.

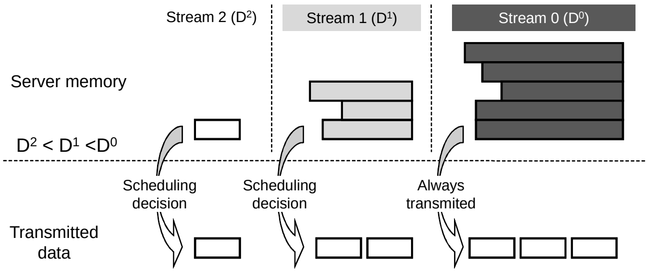

Fig. 2. Scalable scheduling principle ($S = 3$).

Inside the s^{th} substream ($s = 0..S-1$), all packets are characterized by the same D^s value, which is here no longer a distortion measure, but a cost function proportional to the estimated packet importance for the user experienced quality. According to the specific application requirements, this cost function can be a mean square error based distortion measure or an arbitrary combination of estimated distortions. This substream approach provides a way to differentiate NAL unit packets of different regions of the image and to favour regions of interest that would not be treated as priority packets with a classical distortion model.

3.2. Transmission policy

In this section, we propose a time slotted scheduling algorithm in which the scheduling decision frequency is equal to the maximum video refresh frequency. During each time slot, the algorithm can decide to send zero, one or several packets.

Let us focus on a packet belonging to the n^{th} video frame and to the s^{th} substream. Packets are not retransmitted and until a packet is transmitted or discarded because its deadline is passed, it can be examined at each time slot. Inside one stream, packets have the same importance in terms of quality. Therefore, at a given instant, the priority packet of a stream is the head of the line (HOL) packet with the oldest time stamp. Firstly, the algorithm verifies that the remaining channel bandwidth is sufficient to transmit the HOL packet. Let us define Δ_{max} the maximum end to end delay, ts_{hol} the time stamp of the packet, t the current instant and L_p the average size of a packet. To be sent, the packet must comply with a first condition:

$$t + L_n/R_t + L_p/R_t < ts_{hol} + \Delta_{max}, \tag{4}$$

where L_n and R_t are defined in Section 2. In 4, we estimate the arrival time of the packet: $t + L_n/R_t + L_p/R_t$ and we verify that it goes before the deadline: $ts_{hol} + \Delta_{max}$.

If this condition is not verified, the packet is put on the top of the s^{th} substream queue and the algorithm examines the $(s+1)^{th}$ substream. If the condition is verified, the available bitrate is sufficient to allow packet transmission. Then, to decide to send or to delay the packet, the server has to estimate the distortion increase caused by the chosen policy. In order to estimate this distortion, let us define the random process TT (Transmission Time), which represents the time spent by the network to send a packet. Besides, we define δ^s the remaining time to send the HOL packet of substream s before its playout deadline. Then, we consider the network as a single-hop path and assume the traffic to be exponentially distributed and we express the late loss probability:

$$P(TT > \delta^s) = e^{(-\lambda \delta^s)} \tag{5}$$

where $1/\lambda$ is the average transmission time of a packet and δ^s is given by:

$$\delta^s = \Delta_{max} - (t - ts_{hol}). \tag{6}$$

Moreover we can write:

$$1/\lambda = L_n/R_t + L_p/R_t, \tag{7}$$

where the first term corresponds to a delay due to the network congestion and the second term is the transmission delay of the wireless link.

When transmitting a HOL packet, one can expect that $t_{latency}$ will increase due to the capacity of a packet to create congestion. This congestion may potentially penalize future packet transmission. Hence, to take into account this effect, packet queue of the streams with higher priorities (higher D^s values), is considered as a single packet with the time stamp ts_{min} of the oldest packet. Let us denote by $N^0, N^1, .., N^{(s-1)}$ the queue length (number of queuing packets) of the streams with higher priorities. Then we can write the total distortion increase when this aggregation of packets is lost:

$$D^s_{ag} = \sum_{j=0}^{s-1} N^j D^j. \tag{8}$$

Besides, in Eq. 6, ts_{hol} is replaced by ts_{min} and in Eq. 7, L_p is replaced by $\sum_{j=0}^{s-1} N^j L_p$. Next, with this new packet definition, we can express the expected distortion increase:

$$D^s_{exp} = \begin{cases} P(TT > \delta^s)\left\{D^s_{agg} + D^s\right\} & \text{if HOL packet is sent} \\ P(TT > \delta^s) D^s_{agg} + D^s & \text{otherwise.} \end{cases} \tag{9}$$

In Eq. 9, the considered HOL packet is included into the aggregated packet to estimate the expected distortion when the packet is sent, thus L_p and ts_{min} need to be adapted consequently. Hence, the loss probability applied to the aggregated packet is lower when deciding not to send the HOL packet.

Finally, the server chooses the policy that minimizes D^s_{exp}. If the deadline of the packet is not passed and if it is not sent, the packet is put on the top of the s^{th} substream queue and the algorithm examines the $(s+1)^{th}$ substream. In this distortion estimation, packet aggregation is a way to better exploit the SVC structure and the introduced delay dependency between packets allows to better discriminate packets when congestion becomes critical.

4. EXPERIMENTAL RESULTS

To evaluate the efficiency of the proposed approach, the experiments have been conducted using a network simulator provided by the 3GPP video ad-hoc group [11]. In our simulations, all bearers are configured with persistent mode for retransmissions and their bitrates are adjusted using the radio block size and the Transmission Time Interval (TTI) parameters provided by the simulator.

The two parameters: α_{max} and $T_{congest}$ defined in Section 2 have to be set "manually" according to the network characteristics and application requirements. The congestion evaluation depends on the time-out policy applied at the bottleneck entry. Next, we assume that a packet waiting to be transmitted to the radio layer will be discarded after a timeout of one second if no transmission occurs. Based on this assumption, we fix the congestion threshold to half this timeout: $T_{congest} = 500$ms.

In Fig. 3, bitrate variations $R_u(t)$ and $R_t(t)$ are given for two values of the parameter α_{max}, which governs the transmitted bitrate increase αR_u. Hence, we can see that the choice of this parameter mostly influences the beginning of the session when the bitrate is reaching the maximum capacity of the channel. In our experiments we will use $\alpha_{max} = 100\%$, which seems to be a good compromise between stability and rapidity. To evaluate the relevance of our approach, we simulate streaming sessions in a scenario where network conditions become more and more difficult along the time. At the beginning of the session, the channel is set with a bandwidth of 128kbps and a Block Error Rate (BLER) of $3,3\%$. Around

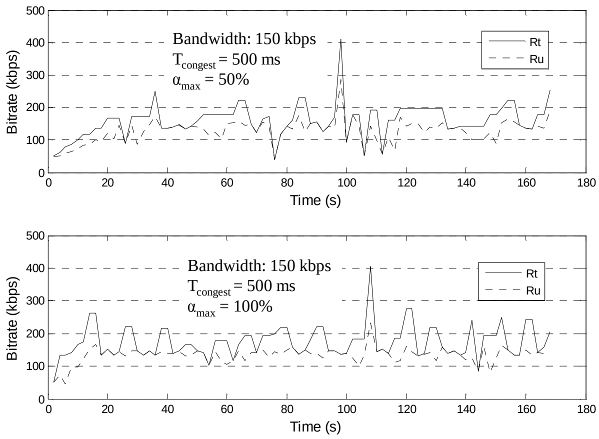

Fig. 3. Transmitted bitrate D_t versus measured user bitrate D_u.

the 30^{th} second, the BLER reaches 10% and the bandwidth falls to 100kbps. During the second part of the session, the BLER remains stable at 10% and the bandwidth reaches a minimum value of 80kbps. In addition, we encode a QCIF video of 75s at 15Hz, which is in fact a concatenation of the well known sequences: COASTGUARD, HALL, MOBILE, STEFAN, BUS, CITY, CONTAINER and FOREMAN. We use one quality refinement layer (one quality layer per substream) and a GOP size of 8. The encoder is configured with constant quantization steps through the video and therefore the bitrate highly depends on the content of the underlying sequence.

Fig. 4 illustrates the ability of our scheduling algorithm to follow channel bitrate variations for two values of the RTCP-RR period. At the beginning of the session, the capacity of the channel is under utilized due to the nature of the first sequences, which only require low bitrates to be transmitted. Next, we can see that with an RTCP period of 1s the bitrate increases faster when the sequence requires more bandwidth. This can be explained by the fact that during the first seconds the server does not know the maximum capacity of the channel and starting around the 20^{th} second, the algorithms increases the bitrate each time it receives an RTCP-RR. In the same way, the server better anticipates the bandwidth decrease when it receives more frequent RTCP-RR.

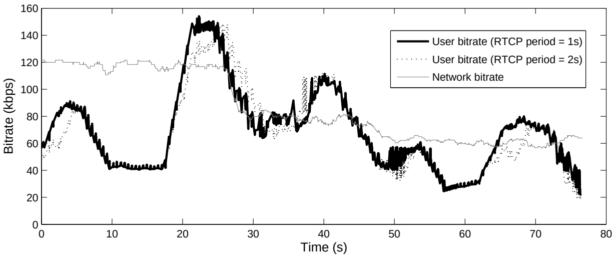

Fig. 4. Bitrate adaptation with two substreams ($q = 0$ and $q = 1$).

To validate our proposed approach, we compare the PSNR values of the decoded video transmitted with three different methods. The first one is a simple transmission of the video encoded with a target bitrate of 100kbps and without any feedback based controls. For the two others we use the knowledge of the network derived from RTCP-RR (receiving period of 1s) to infer the scheduling policy. In one case, a deadline control is applied before sending a packet without considering further dependencies with others packet delays. In the other case we use the optimized congestion control algorithm described in Section 3. Fig. 5 illustrates the ability of our long term feedback driven scheduling algorithm to face a drastic deterioration of network conditions. Moreover, during the transition states, the coupling introduced between packet transmission cost estimation allows to use the available resource more efficiently, leading to an average gain of 0.26dB (on the concatenated sequence of 75s, under the transmission conditions previously described), as presented in Tab. 1.

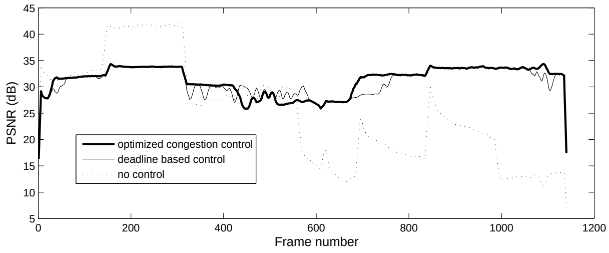

Fig. 5. Comparison of Y-PSNR variations between the three transmission methods.

Transmission method	Average PSNR
Opt. Congest. control	31.34 dB
Deadline based control	31.08 dB
No control	25.32 dB

Table 1. Average PSNR following transmission method.

5. CONCLUSION AND FUTURE WORKS

This paper proposes a complete framework to optimize video streaming applications over wireless networks. The given solution focuses on the case where the media server only receives RTCP-RR as network feedback information. Considering a report period of about one second, we estimate the reception conditions of the user by distinguishing between two functioning modes: congested and not congested. Based on this model, we develop a scheduling algorithm which exploits the SVC structure and assumes a packet delay dependency in order to minimize the risk that a packet leads to penalizing congestion. Experimental results validate our approach, showing the efficiency of the algorithm to face network condition degradation. Future work will tackle a more sophisticated distortion model including psycho-visual perception criteria in order to better discriminate packet importance when choosing an optimal transmission policy.

6. REFERENCES

[1] N. Baldo, U. Horn, M. Kampmann, and F. Hartung, "RTCP feedback based transmission rate control for 3G wireless multimedia streaming," *IEEE Int. Symp. Personal, Indoor and Mobile Radio Com.*, vol. 3, pp. 1817–1821 Vol.3, Sept. 2004.

[2] P. Chou and M. Zhourong, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. on Multimedia*, vol. 8, no. 2, pp. 390–404, April 2006.

[3] J. Chakareski and P. Frossard, "Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources," *IEEE Trans. on Multimedia*, vol. 8, no. 2, pp. 207–218, April 2006.

[4] E. Maani, P. V. Pahalawatta, R. Berry, T. N. Pappas, and A. K. Katsaggelos, "Resource allocation for downlink multiuser video transmission over wireless lossy networks," *IEEE Trans. on Image Proc.*, vol. 17, no. 9, pp. 1663–1671, Sept. 2008.

[5] M. Stoufs, A. Munteanu, P. Schelkens, and J. Cornelis, "Optimal joint source-channel coding using unequal error protection for the scalable extension of H.264/MPEG-4 AVC," *IEEE Int. Conf. on Image Proc.*, vol. 6, pp. 517–520, San Antonio, USA, Oct. 2007.

[6] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable H.264/MPEG4-AVC extension," *IEEE Int. Conf. on Image Proc.*, pp. 161–164, Atlanta, USA, Oct 2006.

[7] D. Tian, X. Li, G. Al-Regib, Y. Altunbasak, and J. R. Jackson, "Optimal packet scheduling for wireless video streaming with error-prone feedback," *Wireless communications and networking conference*, vol. 2, pp. 1287–1292, Atlanta, USA, March 2004.

[8] N. Tizon and B. Pesquet-Popescu, "Scalable and media aware adaptive video streaming over wireless networks," *EURASIP Journal on Advances in Signal Processing*, 2008.

[9] E. Setton and B. Girod, "Congestion-distortion optimized scheduling of video over a bottleneck link," *IEEE Workshop on Multimedia Signal Proc.*, pp. 179–182, Siena, Italy, Sept. 2004.

[10] S. Wenger, N. Sato, C. Burmeister, and J. Rey, "Extended RTP profile for real-time transport control protocol (RTCP)-based feedback," *RFC4585*, July 2006.

[11] 3GPP and Siemens, "Software simulator for MBMS streaming over UTRAN and GERAN," *Doc. for proposal, TSG System Aspects Working Group4#36, Tdoc S4-050560*, Sept 2005.