Journal of Zhejiang University SCIENCE B
ISSN 1673-1581
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

Rate-distortion optimized frame dropping and scheduling for multi-user conversational and streaming video

Wei Tu[1], Jacob Chakareski[2], and Eckehard Steinbach[1]

[1]Media Technology Group, Institute of Communication Networks
Munich University of Technology, Munich, Germany
E-mail: {wei.tu, eckehard.steinbach}@tum.de

[2]Ecole Polytechnique Federale de Lausanne (EPFL)
Signal Processing Institute - LTS4, CH-1015 Lausanne
E-mail: jakov.cakareski@epfl.ch

Received Dec. 15, 2005; revision accepted Jan. 27, 2006

Abstract: We propose a Rate-Distortion (RD) optimized strategy for frame-dropping and scheduling of multi-user conversational and streaming videos. We consider a scenario where conversational and streaming videos share the forwarding resources at a network node. Two buffers are setup on the node to temporarily store the packets for these two types of video applications. For streaming video, a big buffer is used as the associated delay constraint of the application is moderate and a very small buffer is used for conversational video to ensure that the forwarding delay of every packet is limited. A scheduler is located behind these two buffers that dynamically assigns transmission slots on the outgoing link to the two buffers. Rate-distortion side information is used to perform RD-optimized frame dropping in case of node overload. Sharing the data rate on the outgoing link between the conversational and the streaming videos is done either based on the fullness of the two associated buffers or on the mean incoming rates of the respective videos. Simulation results show that our proposed RD-optimized frame dropping and scheduling approach provides significant improvements in performance over the popular priority-based random dropping (PRD) technique.

Key words: Rate-distortion optimization, video frame dropping, conversational video, streaming video, distortion matrix, hint tracks, scheduling, resource assignment

doi:10.1631/jzus.2006.B0001 **Document code:** A **CLC number:** R979.1

1. INTRODUCTION

Video packets are transmitted over the Internet using a best-effort service. Therefore, a large number of simultaneous video streams or cross traffic arriving at a network node (e.g. a multimedia gateway) may sometimes exceed the node's forwarding capacity, i.e., the incoming data rate may exceed the outgoing data rate at the node. In this paper, we consider a scenario where M streaming videos and N conversational videos pass through a network node with limited forwarding resources, illustrated in *Fig. 1*. The packets can be temporarily cached in the node's buffer, but if an overload persists, the buffer will overflow and some packets will be lost. Our goal is to improve the overall reconstruction quality over all streams at their respective receivers for a given forwarding resource R_{out}, which is the data rate on the outgoing link at the node.

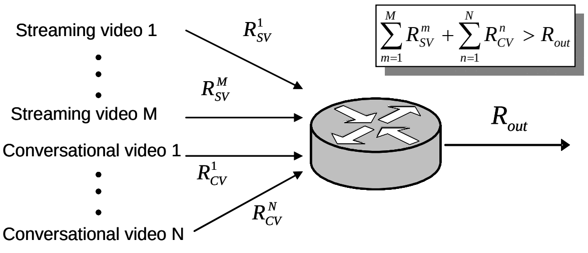

Fig. 1: A network node with M incoming streaming videos and N conversational videos that all have to be sent out via the same outgoing link.

For video applications, transcoding [1] or pruning [2] of the packetized video stream can be employed to adapt the associated source rate to the available transmission rate. Transcoding is computationally expensive and not suitable for a node that has

This work has been supported by DFG grant STE 1093/1-1.

to rapidly forward packets of many different users. On the other hand, video pruning by random frame dropping may have a dramatic influence on the reconstructed video quality. In [3][8][9], static priority labels for I, P and B frames are used to perform priority-based random dropping (PRD) for streaming video.

Priority-based early random dropping (PRED) [12] improves PRD by starting to randomly drop some frames with lower priority at certain predefined buffer fullness levels. However, static priority labels can not accurately describe the importance of the frames. For example, the first P frame in a Group of Pictures (GOP) is in most cases much more important than the last P frame, although they belong to the same class of frames. For conversational video the typical encoding structure is an initial I frame followed by all P frames. All P frames get the same label and there is no priority difference between different frames, while the loss of different P frames has different impact on the reconstruction quality [4].

Jointly optimized multi-user video frame dropping has been proposed in [5] and [7]. In [7], an RD-optimized frame dropping strategy for multiple users is introduced, but in that work, only streaming video is considered. In [5], Hint Tracks (HT) [4] are used for RD-optimized frame dropping for multiple users. Although the encoded videos have an IPPPPP structure, which is normally used for conversational video, the large dropping decision window seems not suitable for applications with tight delay constraints. However, as shown in this work, the idea of HT can be extended to provide RD side information for conversational video.

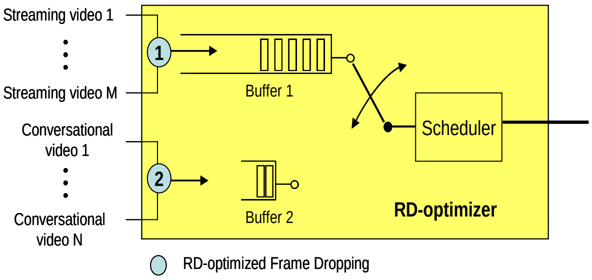

Fig. 2 Structure of the RD-optimizer

In this work, we propose an RD-optimized video frame dropping and scheduling approach for the scenario illustrated in *Fig. 1*. As shown in *Fig. 2*, our RD-optimizer performs two independent dropping decisions for streaming video and conversational video. The surviving frames are stored in two independent buffers. The buffer for conversational video is relatively small in order to limit the forwarding delay. The buffer for streaming video is larger as a result of the moderate delay requirement. A scheduler is located behind the two buffers, which dynamically assigns the shared resource to the two buffers. The resource assignment in our work depends either on the fullness of the individual buffers or on the mean incoming traffic rates.

The rest of the paper is organized as follows. Section 2 introduces the RD-optimized frame dropping strategy and the dynamic resource assignment. Section 3 presents simulation results that show the improvements achieved by our proposed RD-optimized frame dropping and scheduling approach compared with priority-based dropping. Conclusions are presented in Section 4.

2. RD-OPTIMIZED FRAME DROPPING STRATEGY

As shown in *Fig. 2*, our RD-optimizer performs two independent RD-optimized frame dropping decisions and stores the surviving frames in two separate buffers. In this section, we first consider RD-optimized frame dropping for streaming video using two different types of RD side information proposed previously. The first type is the Distortion Matrix (DM), introduced in [7] and the second type is the Hint Tracks described in [4]. Then, in Section 2.2 different frame dropping strategies are proposed for streaming and conversational videos, respectively. Finally, two approaches for dynamic resource assignment are presented in Section 2.3.

2.1 Distortion information for streaming video

2.1.1 Distortion Matrix (DM)

The Distortion Matrix proposed in [7] allows us to calculate the distortion caused by dropping frames in a GOP structured video stream. When calculating the distortion, we assume that a simple copy previous frame error concealment scheme is used by the decoder. Once a specific P frame or I frame is lost, all depending frames in this GOP are replaced with the latest successfully decoded frame. The additional distortion for a particular dropping pattern is the sum of the individual frame distortions of the concealed pictures. The Distortion Matrix for a GOP with IB$_1$B$_2$P$_1$B$_3$B$_4$P$_2$B$_5$B$_6$ structure is given as follows:

$$\begin{array}{r} R: \\ I: \\ P_1: \\ P_2: \\ B_1: \\ B_3: \\ B_5: \end{array}\begin{bmatrix} D_I^R & D_{B_1}^R & D_{B_2}^R & D_{P_1}^R & D_{B_3}^R & D_{B_4}^R & D_{P_2}^R & D_{B_5}^R & D_{B_6}^R \\ / & D_{B_1}^I & D_{B_2}^I & D_{P_1}^I & D_{B_3}^I & D_{B_4}^I & D_{P_2}^I & D_{B_5}^I & D_{B_6}^I \\ / & / & / & / & D_{B_3}^{P_1} & D_{B_4}^{P_1} & D_{P_2}^{P_1} & D_{B_5}^{P_1} & D_{B_6}^{P_1} \\ / & / & / & / & / & / & / & D_{B_5}^{P_2} & D_{B_6}^{P_2} \\ / & / & D_{B_2}^{B_1} & / & / & / & / & / & / \\ / & / & / & / & / & D_{B_4}^{B_3} & / & / & / \\ / & / & / & / & / & / & / & / & D_{B_6}^{B_5} \end{bmatrix} \tag{1}$$

where $D_{F_{loss}}^{F_{rep}}$ are the MSE values observed when replacing frame F_{loss} by F_{rep} as part of the concealment strategy. The column left to the distortion matrix shows the replacement frame F_{rep} for every row of the matrix. For instance, $D_{B_1}^{I}$ represents the additional reconstruction distortion if the first B frame of the GOP is lost and therefore replaced by the I frame of that GOP. R is a frame from the previous GOP that is used as a replacement for all frames in the current GOP if the I frame of the current GOP is lost.

The number of entries of the distortion matrix can be calculated as follows

$$N_{entries} = \frac{1}{2} L\left(3 + \frac{L}{N_B + 1}\right), \tag{2}$$

where L is the length of the GOP, and N_B is the number of B frames between two P or I frames. Given this matrix, our RD-optimized frame dropping strategy for streaming video chooses between four possible dropping decisions that can be made for each stream: drop I frame, drop P frame, drop B frame and drop nothing. As part of our dropping decision, all depending frames in the same GOP are also dropped. For example, if we decide to drop the I frame of a GOP, this involves dropping all other frames from the GOP. Also, if we decide to drop a P frame, this involves dropping all depending B and P frames of this GOP. Although many possible dropping choices are available, previous dropping decisions and also the position of the frames can limit the computational complexity as stated in [7].

When we calculate the DM, simple copy previous frame error concealment is assumed and it is also used as the error concealment scheme at the decoder in [7]. Although the actual error concealment scheme might be more sophisticated, our proposed DM still accurately represents the reconstruction distortion for the scheme in [7] as a result of dropping all dependent frames.

2.1.2 Hint Tracks (HT)

RD Hint Tracks, proposed in [4][10][11], are measured by feeding a specific loss pattern to the decoder and summing up the resulting increase in MSE over all affected frames of the video sequence. For streaming video with an IBPBP… GOP structure, two frames that belong to two different GOPs can be considered independently and distortion can then be calculated individually. Given that the k^{th} frame in display order in a GOP is a P or I frame and is lost, the dependent B and P frames are not dropped. Instead, the decoder performs error concealment for the dropped frame and keeps on decoding. At the beginning of the next GOP, the error propagation terminates, as shown in *Fig. 3*. For B frames, as they are not used as reference frames, there is no error propagation to other frames.

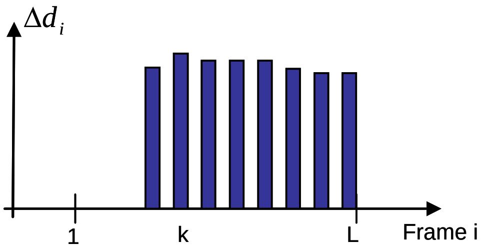

Fig. 3 Error propagation for a single lost frame k in a GOP of L frames and IBPBP… structure.

The estimated distortion for P or I frames is calculated as follows

$$D^0(k) = \sum_{i=k-N_B}^{L} \Delta d_i \,. \tag{3}$$

L is the length of the GOP and N_B is the number of B frames between two I and P frames. Here i starts from $k - N_B$ because the N_B frames before the k^{th} frame in the GOP are B frames, which use the lost frame k as a reference frame. For example, $N_B = 1$ in *Fig. 3*. D^0 represents the distortion under the assumption that each frame loss is independent. However, it is close to the real distortion only when the loss rate is very low and the packet loss process is not bursty in nature. For more accurate distortion estimation, we have to use all D^0 D^1…D^{L-1}, which calculate the additional distortion by dropping the current frame with the knowledge of which of the previous frames in the same GOP have been dropped. However, the computational complexity and storage cost of these higher order HTs are quite high. For D^1, $L*(L-1)/2$ entries are needed for the video stream with GOP length of L, and for higher order HTs, the number of entries increases polynomially. At high packet loss rates or when successive frame loss happens, HT with D^0 or D^1 leads to false distortion estimation and hence suboptimal dropping decisions. The performance of a HT based approach for frame dropping depends on the number of frames that are considered jointly when the optimization is performed. In our case, a window of size W can be used so that at every moment W frames from each stream are taken into account for the optimization.

Our experimental results in Section 3.1 show that RD-optimized frame dropping using DM typically outperforms RD-optimized frame dropping using low order HT for streaming video. Therefore, we describe the RD-optimized frame dropping for streaming video in the following using DM as side information. For conversational video, the DM cannot be used and therefore we employ the HT approach for this video application.

2.2 RD-optimized video frame dropping

We propose in this work that the dropping decision for streaming video is determined by minimizing a Lagrange cost function

$$J_p(i) = \sum_{m=1}^{M} \Delta D_p^m(i) - \lambda(i) \sum_{m=1}^{M} \Delta R_p^m(i), \quad (4)$$

where $\sum_{m=1}^{M} \Delta D_p^m(i)$ and $\sum_{m=1}^{M} \Delta R_p^m(i)$ are extracted from the DM and represent the sum of the additional distortion and rate saving of all M streaming videos at frame i given dropping pattern p. $\lambda(i) > 0$ is the Lagrange parameter at frame i, which can be determined from the current buffer fullness [7]. When the buffer is lightly loaded, a small λ is used to give more importance to the incurred distortion. But rate saving is more important when the buffer is highly loaded. For streaming video, as the importance of current and future frames are known, some less important frames can be early dropped to free space for more important future frames.

For conversational video, the importance of future frames is unknown, which makes it unnecessary to make dropping decisions before the buffer is full. As the buffer size for conversational video is relatively small, we put all new frames to the tail of the buffer queue if there is enough space left. Otherwise, we compare the distortion per-bit utility [4] for frame i defined as the ratio $\Delta D^n(i)/R^n(i)$ with the corresponding utility values for all frames in the buffer. Here $\Delta D^n(i)$ is the additional reconstruction distortion that is incurred when frame i from stream n is dropped and $R^n(i)$ is the size in bytes of this frame. The frames in the buffer with lower utility than the new incoming frame will be first marked to be dropped. If the buffer space released by dropping these frames is enough to put in the new frame, they are physically dropped from the buffer. On the other hand, if the released space is not enough to hold the new frame, it means the new frame is either too big or is not important enough for the reconstruction quality of the corresponding stream. Then this new frame is dropped and the marked frames in the buffer are recovered. Note that this approach is equivalent to that taken in [4] for creating priorities among frames in a transmission window at a streaming server.

2.3 Scheduling for streaming and real-time video

Two separate buffers are employed to limit the additional delay experienced by the conversational video streams, as explained earlier. Compressed video has variable bit-rate, and hence fixed resource assignment may sometimes waste the resources, and sometimes lead to dropping some frames that could be avoided. With a dynamic resource assignment in place, the multiplexing of the multiple streams decreases the variation of the bit-rate and provides for more efficient resource utilization. Here, we propose two schemes for dynamic assignment of the data rate on the outgoing link. The first scheduling scheme is based on the short-term mean rates of the incoming streaming and conversational videos. The second scheme takes the buffer fullness level into account to avoid buffer overflow.

2.3.1 Short-term mean rate based scheduling

Compressed video streams are typically VBR (Variable Bit Rate), so when the outgoing link provides a transmission rate equal to the mean rate of the video stream, most likely some packets will be dropped if there is only a very small buffer at the node. But if we can perform the assignment adaptively following the variability of the stream's bit-rate, the resource can be more efficiently used. Without the knowledge of the size of future frames for conversational video, we can only make an estimation of the future bit-rate with the knowledge of the rate history. Here, we present a straight forward way to accommodate for this. We take *F* past frames from each stream as an estimation window. The current resource assignment is then calculated as follows:

$$r_{SV}^{i}=\sum_{j=1}^{M}\sum_{k=i-F-1}^{i-1}R_k^j \text{ and } r_{CV}^{i}=\sum_{j=1}^{N}\sum_{k=i-F-1}^{i-1}R_k^j \qquad (5)$$

$$\begin{aligned} S_{SV}^{i}&=R_{out}\times\frac{r_{SV}^{i}}{r_{CV}^{i}+r_{SV}^{i}} \\ S_{CV}^{i}&=R_{out}-S_{SV}^{i} \end{aligned} \qquad (6)$$

Here, r_{SV}^{i} and r_{CV}^{i} are the sum of bytes from the previous *F* frames of *M* streaming videos and *N* conversational videos. S_{SV}^{i} and S_{CV}^{i} represent the assigned transmission rate to the two buffers. R_{out} is the total transmission rate on the outgoing link and it is assumed to be constant during the whole transmission. With the same formula (6), dynamic resource assignment for variable data rate on the outgoing link rate can also be calculated.

2.3.2 Buffer fullness based scheduling

Buffer fullness based scheduling is an efficient way for the scheduler to avoid buffer overflow. When the buffer is heavily loaded, it means the incoming rate is bigger than the assigned service rate and therefore new incoming frames are likely to be dropped. In this case, a large portion of the outlink rate should be assigned to this buffer. When the buffer is lightly loaded, the buffer can still hold some new coming frames and more transmission slots can be given to the other buffer. The transmission rate assigned to the streaming videos at frame *i* can be calculated with (7) and (8) and the remaining transmission capacity is assigned to the conversational videos.

$$\begin{aligned} r_{SV}^{i}&=\frac{1}{M\cdot(i-1)}\cdot\sum_{j=1}^{M}\sum_{k=1}^{i-1}R_k^j \\ r_{CV}^{i}&=\frac{1}{N\cdot(i-1)}\cdot\sum_{j=1}^{N}\sum_{k=1}^{i-1}R_k^j \end{aligned} \qquad (7)$$

$$S_{SV}^{i}=R_{out}\times\frac{B_{SV}^{i}}{B_{SV}^{i}+B_{CV}^{i}}\times\frac{r_{SV}^{i}}{r_{CV}^{i}} \qquad (8)$$

r_{SV}^{i} and r_{CV}^{i} are respectively the mean incoming rates of the streaming videos and the conversational videos from the beginning till frame *i-1*. B_{SV}^{i} and B_{CV}^{i} denote the percentage of buffer load at the time instance when the *i*th frames of every stream arrives at the node.

3. SIMULATION RESULTS

In this section, we first compare the performance of RD-optimized frame dropping for streaming video when using DM or HT-D^0 as introduced in Section 2.1. Then, for streaming and conversational videos we investigate the improvement in average reconstruction quality that can be achieved by using the proposed RD-optimizer compared with priority-based random dropping (PRD). Finally, to show the influence of the delay constraint on the reconstruction quality of the conversational videos, we examine different sizes for the corresponding buffer.

In our simulations, we assume that four streaming videos are pre-encoded and four conversational videos are offline encoded with the H.264 codec [6]. They arrive at an active network node and have to be sent out on the same outgoing link. Long test sequences are generated by concatenating several classic short test sequences. For simplicity, we always put the first frame of the new short sequence at the beginning of a new GOP for streaming video, which means that the last several frames in the short sequence may be cut out if they are fewer than one GOP length.

In *Table 1,* the first row below the name of the short sequences shows their length in number of frames. The numbers in the rows of the long test sequ-

Test sequence	Carphone	Claire	Foreman	Grandma	Miss_ America	Mother & Daughter	Salesman	Suzie
frames	380	270	400	300	150	320	220	150
SV_1_20	5	1,4	3		2	6		
SV_2_22		5,6	4	3	1	2		
SV_3_24		4	2	6	3	1,5		
SV_4_26		3		4	1	2,6	5	
CV_1	1	3	4			5		2
CV_2	3		1	2,6	5			4
CV_3		2		3,6		1,5	4	
CV_4	6	3			2	4	1,5	

Table 1 Construction of the test sequences

Name	SV_1	SV_2	SV_3	SV_4	CV_1	CV_2	CV_3	CV_4	Sum/Avg
mean rate (Kbps)	92.44	67.99	69.55	50.60	122.07	119.06	67.81	116.42	705.94
mean PSNR (dB)	38.63	38.32	38.10	38.24	37.57	37.26	37.36	37.69	37.90

Table 2 Characteristics of the test sequences

ences represent the order that the short sequences are concatenated, i.e., the first frame of *Suzie* is just after the last frame of *Carphone* for CV_1. The test sequences are named SV_X_YY for streaming video, where YY stands for the length of the GOP and X is just the index of the video. The number of B frames between two P or I frames is set to be 1 in the encodings.

Table 2 summarizes the main characteristics of the eight videos. A frame rate of 25 frames/sec is assumed. For conversational video, in order to increase the error resilience, we insert one INTRA coded row of MBs every two frames, which results in an INTRA update period of 18 frames. As the very first frame of all test sequences is an I-frame, to avoid the loss of the first I frames, we assume that they have gone through the node and all dropping decisions are made after the arrival of the second frames of each stream. The respective sizes of the buffers for streaming and conversational videos are set to be 32KByte and 5KByte in our simulations.

3.1 DM compared with HT-D^0 for streaming video

In this simulation, we compare the performance of DM with that of HT-D^0 when they are employed for RD-optimized frame dropping of streaming video. Both types of side information are computed when the streaming videos are encoded off-line. The four streaming video test sequences described in Table 1 are used.

We also compare the RD-optimized frame dropping with PRED for streaming video, which has been introduced in Section 1. In our simulation, the two thresholds T_1 and T_2 for B and P frames are set to be 70% and 90% of the full buffer load, respectively.

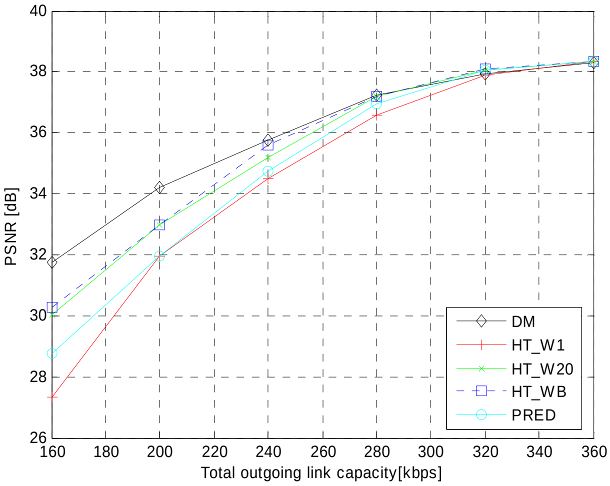

Fig. 4 RD-optimized frame dropping for streaming video using DM and HT-D^0

Fig. 4 shows the average reconstruction quality (mean luminance PSNR) of the four streaming videos. DM and HT-D^0 are used for the RD-optimized frame dropping. In this simulation, DM means that the distortion matrix in combination with (4) is used for RD-optimized dropping as described in [7]. HT_W1 uses HT as shown in *Fig. 3* instead of DM when the optimization is done. The same cost function and same way of calculating λ are used, but the frame dropping strategy with HT makes the dropping decision only for the currently incoming frames, which means the decision window size is *W=1*. At high outgoing rate, the number of dropped frames is low, so HT-D^0 accurately represents the true distortion. At low outgoing link rate, the distortion estimation of the HT-D^0 is no longer correct. Because of the wrong estimation of the reconstruction distortion, it performs even worse than PRED at very low outgoing link rates.

A small window for HT-D^0 based dropping may lead to an unfair comparison. Because of that, the dropping decision using HT-D^0 could be done with a decision window of size *W*. The optimizer takes the side information of future frames in the decision window to minimize the cost function (4). Dropping decision can be made for all frames in the decision window, but only the current frames are put into the buffer if they are decided not to be dropped. Surviving frames in the decision window can always be decided to be dropped in later dropping procedures before they are finally put into the buffer. Here, the decision window size should be properly selected. A too small window leads to less optimal, while too big window includes uncorrelated frames and increases the computational complexity. Here, we set the decision window size to be 20 frames/video and its performance shows as HT_W20 in *Fig. 4*. HT_WB is achieved under the assumption that even frames in the buffer could still be updated, which might be unrealistic for large buffer, but it gives an upper bound on the achievable performance for HT-based frame dropping strategies. However, this upper bound is very close to HT_W20, because at high rates, updates are rarely needed, while at low rates, HT cannot provide accurate distortion estimation. From the simulation results, we can see that with the big decision window, more than 2dB improvement for low link rates is observed when compared with HT_W1, but still inferior performance compared with RD-optimized frame dropping using DM.

3.2 Performance comparison of RD-optimized frame dropping and PRED/PRD

In this experiment, we compare our proposed scheme with priority-based random early dropping (PRED) for streaming video as described in the last simulation and conventional priority-based random dropping (PRD) for conversational video. Because of the big difference of their delay constraint, we employ separate buffers for the two different kinds of video streams as shown in *Fig. 2*. The PRD always drops a frame when the buffer is not able to hold new incoming frames. For conversational video, most of the frames are P frames and there is no static priority difference between them, so when multiple frames are coming at the same time instance, a simple round robin scheme is used to check whether a frame can be put into the buffer.

Our proposed optimizer uses the DM for streaming video and HT for conversational video. In this simulation, HT-D^0 is used, because it is shown in [10] that HT-D^0 has a performance close to HT-D^1 at a lower computational complexity. As the future RD information of future frames of the conversational video is unknown, it is impossible to pre-measure the HT. Therefore, we use the model proposed in [11] to estimate the total distortion $D(i)$, which is the sum of the distortion when frame i is lost plus the distortion due to error propagation over successive frames.

$$D(i) = \sum_{l=0}^{M} MSE(i+l) \tag{9}$$

$$MSE(i+l) = \begin{cases} MSE(i)\cdot r^{l}\cdot(1-l/M) & \textit{for } 0 \le l \le M; \\ 0 & \textit{otherwise.} \end{cases} \tag{10}$$

M is the INTRA update period, which is 18 frames in this simulation. *l* indicates the distance between the concealed frames and the lost frame. $MSE(i)$ is the MSE information sent along with the video stream, representing the reconstruction distortion associated with the loss of frame *i* only, where the missing frame is concealed by copying the previous frame (*i-1*). The attenuation factor r^l with $r \prec 1$ accounts for the effect of spatial filtering and is set to be 0.997. $1-l/M$ accounts for the error reduction due to INTRA update. It is assumed that the error is completely removed by INTRA update after M frames.

Fig. 5 shows the improvements obtained by the RD-optimized video frame dropping and scheduling strategy proposed in this paper. The PSNR values are averaged over the eight video sequences. When the outgoing link rate is larger than the mean incoming rate, the performances of the RD-optimizer and PRED/PRD are close. But there is still a gap at 900kbps, because the small buffer for conversational videos can not hold too many frames during peak rate periods of the video streams. Then the optimized dropping has more opportunities to drop those least important frames even if they have been in the buffer waiting for being sent out.

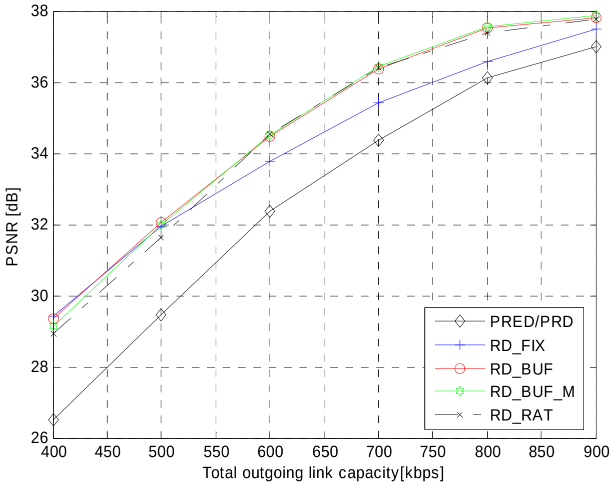

Fig. 5 Performance comparison of the proposed RD-optimized frame dropping scheme and PRED/PRD

At the same time, dynamic resource assignment can also bring some spare transmission slots from the streaming video. When the outgoing rate is smaller than the total traffic rate, a performance gap of around 3 dB can be observed. RD_FIX uses a fixed resource scheduling, while RD_BUF and RD_RAT represent the buffer fullness based and the short term mean rate based scheduling strategies introduced in Section 2.3. At low rate, the performance of these three scheduling strategies is almost the same. At high rate, dynamic resource assignment performs much better, because giving some resource from buffer1 to buffer2 will not influence the quality of streaming video too much, as those resources are mostly spared during the occurrence of the low data rate periods of the video streams. But with fixed resource allocation, these spare resources are wasted, which leads to degraded performance at high outgoing link rates compared with the dynamic resource assignment strategy.

Pre-computed HTs are not available in practice, but here we compute them in order to see if the approximation in (9) and (10) leads to accurate results. Our experiments show that pre-computed HT (RD_BUF_M) for the conversational videos and the approximation (RD_BUF) obtained using (9) and (10) lead to almost identical results. The estimation bias from the model does not affect the results because what we use in this simulation is the relative importance (D/R) between frames, and not absolute distortion values.

3.3 Delay constraint for conversational video

In the above simulations, we assumed that the size of the buffer for conversational video B_{CV} is 5KBytes. This buffer size ensures that in our simulation most of the frames are delayed for less than 3 frame periods. For a different delay constraint, the corresponding size of the buffer has to be adapted. In this section, we investigate the influence of buffer size on average delay that packets suffer in the buffer and the reconstruction quality. In the following experiments, we assume a fixed outgoing rate is assigned to buffer B_{CV} and observe only the average reconstruction quality of the 4 conversational videos.

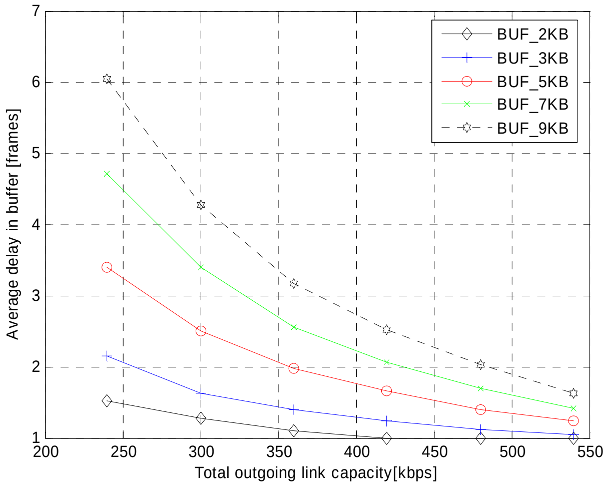

Fig. 6 Average delay with different buffer size for conversational videos

Fig. 6 shows the packet delay in the buffer, which is averaged over the delay of all packets from all conversational videos. As the packets are scheduled at every frame slot, the minimum delay in the buffer for the packets corresponds to one frame slot. At low transmission rates, large buffer introduces significant delay. When the buffer size is 5KBytes, only the average delay at the lowest rate exceeds 3 frames. With the knowledge of the estimated average total bitrate of the conversational videos, it is not difficult to select the proper buffer size to fit the delay constraint.

Average delay increases with increasing buffer size. However, from *Fig. 7* we can see that under the settings of our simulation, when the buffer size B_{CV} at 5 KBytes is compared with that at 7 and 9 KBytes, there is only a small performance difference between them. But when B_{CV} is equal to 3 KBytes, the performance degrades around 1 dB on average. The reason for this is that there are some scene changes in the video stream coded as P frames, however with a high number of intra-encoded macroblocks (MBs), e.g., 99 INTRA MBs. Therefore, corresponding frames are very large. When more than one such scene change frames coincide, the buffer can not hold all of them, which causes large distortion. The error propagates, and either stops at the next scene change frame or is stopped by the periodic INTRA update. When the buffer has a size of 2 KBytes, it has the problem to hold even the incoming 4 frames at every time slot. As we get and send frames at every frame slot in our simulation, even at high outgoing link rates, the PSNR will not increase any further.

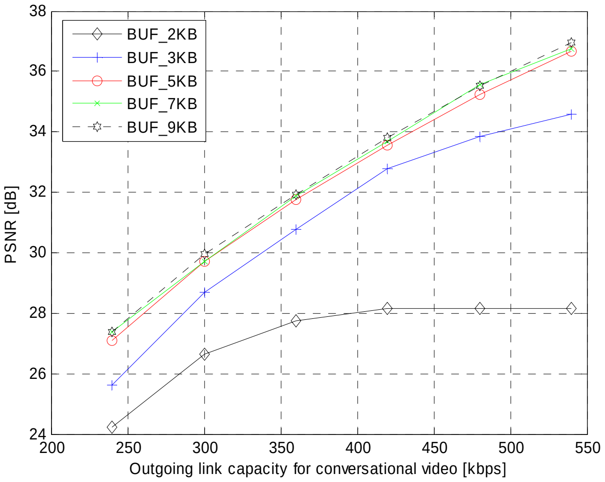

Fig. 7 Performance with different buffer size for conversational videos

4. CONCLUSIONS

We have presented an RD-optimized video frame dropping and scheduling strategy for streaming and conversational video applications that is applied at active network nodes. RD information is extracted when the encoding is performed and sent along with the video streams. The only information the optimizer needs to extract is the current fullness of the buffer and the mean traffic rate of streaming video and conversational video passing through the node. Frame dropping uses RD side information to determine which frames should be dropped in case of heavy network load. Different types of side information in the form of DM and HT are examined for performing RD-optimized frame dropping for streaming video. Mean rate based and buffer-fullness based scheduling strategies are proposed to achieve an optimized dynamic resource assignment. Significant quality improvements are reported when comparing our proposed RD-optimizer to priority-based random frame dropping.

REFERENCES

[1] A. Vetro, C. Christopoulos, and H. Sun, “Video transcoding architectures and techniques: an overview,” Signal Processing Magazine, IEEE, Vol. 20, Issue: 2, pp. 18-29, Mar. 2003.

[2] J. Chakareski and Pascal Frossard, “Low-complexity adaptive streaming via optimized a priori media pruning,” in Proc. Workshop on Multimedia Signal Processing, Shanghai, China: IEEE, Oct./Nov. 2005.

[3] W. Feng, M. Liu, B. Krishnaswami, and A. Prabhudev, “A Priority-Based Technique for the Best-Effort Delivery of Stored Video,” SPIE/IS&T Multimedia Computing and Networking 1999, San Jose, CA, Jan. 1999.

[4] J. Chakareski, J. Apostolopoulos, S. Wee, W. Tan, and B. Girod, “R-D Hint Tracks for Low-Complexity R-D Optimized Video Streaming,” Proc. ICME 2004, Taipei, Taiwan, June 2004.

[5] J. Chakareski and P. Frossard “Rate-Distortion Optimized Packet Scheduling over Bottleneck Links,” Proc. ICME 2005, Amsterdam, Netherlands, July 2005.

[6] JVT H.264/MPEG-4 AVC reference software at http://bs.hhi.de/~suehring/tml/

[7] W. Tu, W. Kellerer, and E. Steinbach, “Rate-Distortion Optimized Video Frame Dropping on Active Network Nodes,” Proc. Packet Video Workshop 2004, Irvine, CA, Dec. 2004.

[8] Y. Lu and K. J. Christensen, “Using Selective Discard to Improve Real-Time Video Quality on an Ethernet Local Area Network,” International Journal of Network Management, Vol. 9, Issue 2, pp. 106 – 117, Mar.–Apr. 1999.

[9] H. Cha, J. Oh, and R. Ha, “Dynamic Frame Dropping for Bandwidth Control in MPEG Streaming System,” Multimedia Tools and Applications, pp. 155-178, vol. 19, no. 2, Feb. 2003.

[10] J. Chakareski, J.G. Apostolopoulos, S. Wee, W. Tan, B. Girod, “Rate-Distortion Hint Tracks for Adaptive Video Streaming,” IEEE Trans. on CSVT, vol. 15, no. 10, Oct. 2005.

[11] Y. Liang, J. G. Apostolopoulos, and B. Girod, “Analysis of Packet Loss for Compressed Video: Does Burst Loss Matter?” Proc. ICASSP 2003, Hongkong, China, April 2003.

[12] R. Mahajan, S. Floyd, and D. Wetherall, “Controlling High-Bandwidth Flows at the Congested Router,” Proc. ICNP2001, Riverside, CA, November 2001.