RATE-DISTORTION OPTIMIZED VIDEO FRAME DROPPING ON ACTIVE NETWORK NODES

Wei Tu[1], Wolfgang Kellerer[2] , and Eckehard Steinbach[1]

[1]Institute of Communication Networks
Media Technology Group
Technische Universität München, Munich, Germany
{wei.tu, eckehard.steinbach}@tum.de
[2]DoCoMo Communication Laboratories Europe GmbH, Munich, Germany
kellerer@docomolab-euro.com

Abstract **- We propose a rate-distortion optimized video frame dropping strategy that can be applied on active network nodes in case of heavy traffic load. Our approach relies on side information that is sent along with the video bit-streams. The side information consists of a rate vector containing the frame size (in bytes) of every picture as well as a distortion matrix describing the reconstruction distortion (in MSE) observed for a Group of Pictures given a certain frame loss pattern. In contrast to QoS-based approaches, our scheme offers larger flexibility and supports dynamic frame importance control. When comparing our rate-distortion optimized dropping concept with priority-based dropping strategies, significantly improved reconstruction quality is observed. Improvements of up to 8dB are obtained for our simulation setup.**

I. INTRODUCTION

Video streaming over the Internet - or any other kind of network where the resources are shared by many users - faces the problem that at one or more points along the path the incoming rate at a network node might be higher than the outgoing rate. This leads to an increasing buffer fullness of that node and eventually to packet loss. For video, if more traffic arrives than the outgoing link supports, the video data either has to be transcoded to lower rate or video packets have to be dropped.

Transcoding is computationally expensive and is not considered further here. Random frame dropping can have a very dramatic influence on the video quality. Here, scalable video offers the opportunity to drop less important parts of the video bit-stream first, which leads to graceful degradation as traffic increases. QoS labeling of the video packets together with priority mechanisms in the network nodes support importance-controlled dropping of the data. The label (or importance) of the packets is decided by the sender before transmission and does not consider the actual transmission situation. This is a disadvantage as the importance of a packet might change along the transmission path. Consider, e.g., a video stream with temporal scalability which has the following Group of Picture structure: IBBPBBP… . If a network node drops a B-frame, the other frames are not affected. If, however, the P-frame after the I-frame is dropped, all following frames (B- and P- frames) up to the next I-frame will be affected as they depend on the dropped frame. Therefore, if we know that the first P-frame has to be dropped, the importance of all following frames changes. Also, as we usually have only a few different importance labels available in the network, different frames with the same label will nevertheless have different influence on the reconstruction quality at the receiver.

A widely investigated approach to deal with varying importance of video frames is rate-distortion (RD) optimization. RD-optimized frame handling has been successfully used in many situations. In [7] for instance, it is used to achieve RD-optimized frame scheduling for a single video stream. RD-optimization for bit allocation between source coding and channel coding is used for instance in [8]. RD-optimization is also state-of-the-art for coding mode selection during encoding [9]. However, those works all concentrate on the video encoding at the sender to choose the best encoding and sending strategy according to the constrained transmission rate and expected loss rate. When RD-optimization is done on some nodes inside the network, side information has to be transmitted along to those nodes [10][11]. [10] considers RD-optimization in a broadcast networking scenario, and both, [10] and [11] do RD-optimization only on a single stream.

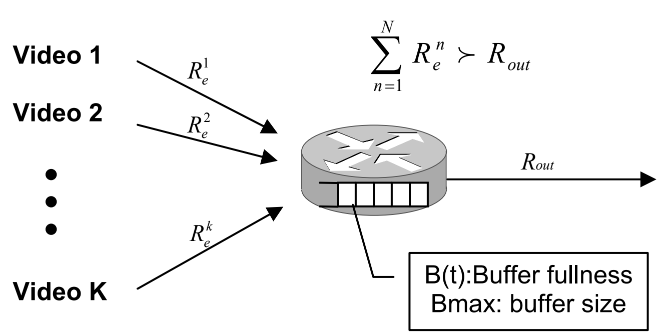

Figure 1: An active network node with K incoming video streams that all have to be sent out via the same outgoing link.

In our work, we use RD-optimization to make an optimized frame dropping decision in case of network congestion for multiple streams simultaneously. We propose in this paper to send a rate vector and a distortion matrix for each stream as side information along with the video bitstream. This side information can be used by active network nodes to dynamically decide in a RD-optimized way which frames of which video stream should be dropped in case of node overload. We propose a Lagrangian cost function that uses the rate vectors and the distortion matrices of all incoming streams together with the current buffer fullness level to find the optimum dropping pattern.

We consider a scenario where K video streams arrive at an active network node and leave the node on the same outgoing link as shown in Fig. 1. An active network node is an entity that has to forward incoming data streams to one or more outgoing links (wireless or wired) and has more computational resources than those required for just forwarding the packets. Examples are programmable or active routers, proxies, or gateways [2], the base station of a cellular network, or the cable head-end of a cable network.

In [1], a dropping strategy for scalable video is introduced that can be implemented on active routers. In comparison to our work, the dropping decision is not made in a RD-optimized way. In [6], a video frame dropping strategy is proposed that discards all those frames depending on previously dropped frames. Again, the dropping decision does not consider many simultaneous video streams in a RD-optimized way.

This paper is organized as follows. In Section II, the rate vector and distortion matrix are introduced. The RD-optimized frame dropping strategy is described in Section III. Section IV presents simulation results that show the improvements achieved by our proposal compared with priority-based dropping.

II. RATE VECTOR AND DISTORTION MATRIX

The frame dropping strategy proposed in this paper relies on side information that is sent along with the video bit-streams. We assume in the following that the video is organized in Group of Pictures (GOP) and the active network nodes know the frame structure of a GOP. This information can either be signaled along with the bit-stream or can be inferred from previous GOPs. The GOP structure of video k is described by the GOP length L^k and the number of B-frames B^k in between two I- or P-frames. As an example, let's consider $L^k = 9$ and $B^k = 2$ which leads to the GOP structure $IB_1B_2P_1B_3B_4P_2B_5B_6$.

The rate vector consists of the frame size of every frame in the GOP. The frame size of frame n in video k is denoted as $R_e^k(n)$ and can be extracted during the encoding process at the sender. The mean encoding rate is therefore given as the average of $R_e^k(n)$ over the entire video sequence of length N_k

$$R_e^k = \frac{f^k}{N_k}\sum_{n=1}^{N_k} R_e^k(n) \tag{1}$$

with f^k being the frame rate of video k.

The distortion matrix consists of the reconstruction distortion values observed when replacing a lost frame by a preceding frame. The reconstruction distortion of frame n of video k is given as $D^k(n)$. The mean distortion at the decoder therefore is given as

$$D^k = \frac{1}{N_k}\sum_{n=1}^{N_k} D^k(n) \quad . \tag{2}$$

If every video frame is received correctly, the frame distortion $D^k(n)$ corresponds to the encoding distortion $D_e^k(n)$ which is caused by quantization at the encoder. In case frames are missing for decoding, the frame distortion will be larger than the encoding distortion. In this work we assume that in case of frame loss, the decoder applies a concealment strategy where the most recent decoded frame is displayed instead of the lost frame. All frames depending on a lost frame are considered to be lost as well. The distortion matrix contains all distortion values that are necessary to infer the expected distortion at the decoder in case of a frame drop given the aforementioned concealment strategy. The following example shows a distortion matrix for a GOP with $L^k = 9$ and $B^k = 2$.

$$\begin{array}{r} R: \\ I: \\ P_1: \\ P_2: \\ B_1: \\ B_3: \\ B_5: \end{array}\begin{bmatrix} D_I^R & D_{B_1}^R & D_{B_2}^R & D_{P_1}^R & D_{B_3}^R & D_{B_4}^R & D_{P_2}^R & D_{B_5}^R & D_{B_6}^R \\ / & D_{B_1}^I & D_{B_2}^I & D_{P_1}^I & D_{B_3}^I & D_{B_4}^I & D_{P_2}^I & D_{B_5}^I & D_{B_6}^I \\ / & / & / & / & D_{B_3}^{P_1} & D_{B_4}^{P_1} & D_{P_2}^{P_1} & D_{B_5}^{P_1} & D_{B_6}^{P_1} \\ / & / & / & / & / & / & / & D_{B_5}^{P_2} & D_{B_6}^{P_2} \\ / & / & D_{B_2}^{B_1} & / & / & / & / & / & / \\ / & / & / & / & / & D_{B_4}^{B_3} & / & / & / \\ / & / & / & / & / & / & / & / & D_{B_6}^{B_5} \end{bmatrix} \tag{3}$$

The entries in the distortion matrix $D_{F_{loss}}^{F_{rep}}$ 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. From this matrix, the resulting distortion for any possible loss pattern can be determined. The total distortion for the GOP is computed as the sum of the individual frame loss distortions. This matrix can be determined during the encoding of the video. The B-frames B_2, B_4, and B_6 will never be used as a replacement frame for any other frame in the GOP and therefore do not show up in the distortion matrix. The number of columns of the distortion matrix corresponds to the GOP length L^k. The number of relevant entries into the distortion matrix can be determined as

$$L^k + \left(L^k - 1\right) + \sum_{i=0}^{\frac{L^k}{B^k+1}-1} \left(B^k + i\left(B^k + 1\right)\right) \quad (4)$$

which can be simplified to

$$\frac{1}{2} L^k \left(3 + \frac{L^k}{B^k + 1}\right) \quad . \quad (5)$$

The side information has to be transmitted along with the video bitstream. For the rate vector, the size of a frame is signaled using two bytes. The entries of the distortion matrix are also signaled with two bytes, which results in $L^k \cdot \left(5 + \frac{L^k}{B^k + 1}\right)$ bytes of side information for every GOP. The larger the size of the GOP, the more side information is needed. A detailed calculation of the side information overhead is stated in Section IV.

III. FRAME DROPPING STRATEGY

We consider the scenario where K video streams arrive at an active network node and leave the node on the same outgoing link. This outgoing link has a transmission rate R_{out} . The outgoing link has a link buffer of size $B_{\max}$ and the current buffer fullness is denoted as $B(t)$. Our dropping strategy is based on the current buffer fullness. If the buffer is empty, no frames should be dropped. When the buffer fills up, those frames should be dropped that have the least impact on the perceived quality at the receiver. The decision which frames to drop has to be jointly made for all video streams. Given the rate vector and the distortion matrix described above, the active network node can perform RD-optimized frame dropping. For this, the node determines how full the buffer currently is and minimizes a Lagrangian cost function that determines the optimum dropping pattern

$$J_p(n) = \sum_{k=1}^{K} \Delta D_p^k(n) - \lambda(n) \sum_{k=1}^{K} \Delta R_p^k(n) \quad (6)$$

where $\Delta D_p^k(n)$ is the additional distortion introduced in video k for dropping pattern p and $\Delta R_p^k(n)$ is the number of bytes saved for dropping pattern p . For simplicity, we replace the continuous time t by the frame index n of the video sequences, which means that dropping decisions will only be made at multiples of the frame duration.

In case the streams have different frame rates, the dropping decision can be made synchronized to the stream with the highest frame rate. Another approach would be to collect a small number of frames from all incoming streams and then perform the decision. This approach, however, would introduce a small amount of additional delay.

If the current frame that arrives at the active node is an I-frame, we can either drop this frame or send it to the outgoing link. If we drop it, this means that all following P- and B-frames cannot be decoded and have to be dropped also. This dropping strategy leads to a significant increase in distortion for this GOP but at the same time allows us to reduce the sending rate to 0 for this GOP. If we do not drop the I-frame at time n , we can still decide to drop all following P-frames. This will lead to reduced distortion but also the rate saving will be smaller. If we decide not to drop the following P-frames, we could decide to drop all B-frames. Again, the additional distortion will be reduced but also the rate saving will be even smaller. So if the current incoming frame is an I-frame, there is a total of 4 dropping choices {I,P,B,N}, where N stands for drop nothing. If the current frame is a P-frame, the choices are reduced to {P,B,N}. If the current frame is a B-frame the choices are also {P,B,N}. Please note that in this case the P stands for the next P-frame that is transmitted after the current B-frame. One could imagine other dropping patterns for the B-frames. As the rate saving, however, is typically very small when dropping a single B-frame, we assume in the following that dropping B-frames always means dropping this B-frame and all following B-frames in the same GOP. Similarly, dropping P-frames always means dropping all P- and B-frames up to the next I-frame and dropping one I-frame results in dropping the entire GOP. If we denote the number of possible dropping patterns at time n for video k as $A^k(n)$ then for K videos we get $P(n) = \prod_{k=1}^{K} A^k(n)$ different dropping patterns. One of the dropping patterns will minimize (6). This pattern represents the optimum dropping strategy at time n .

In order to perform this minimization, we have to determine a reasonable value for the Lagrangian multiplier $\lambda(n)$ in (6). In this work, we determine $\lambda(n)$ as a function of buffer fullness $B(n)$. If the buffer is empty, we certainly do not want to drop any video frames. This has to be reflected by an appropriate choice of $\lambda(n)$. If the buffer is full, $\lambda(n)$ should be selected such that all possible frames are dropped as the enqueuing of these frames in the outlink buffer would fail anyway. In order to determine appropriate values for $\lambda(n)$ at any buffer level, we define a minimum buffer fullness $B_{\min}$, below which no dropping should happen and a maximum buffer fullness $B_{\max}$ over which all frames are dropped. The two buffer fullness values $B_{\min}$, $B_{\max}$ and the corresponding dropping strategies lead to two extreme values for the Lagrange multiplier $\lambda_{\min}(n)$ and $\lambda_{\max}(n)$. The values for $\lambda(n)$ between $B_{\min}$ and $B_{\max}$ are interpolated. We consider and compare two different interpolation schemes for $\lambda(n)$ in this work.

3.1 Linear Interpolation (LI)

Fig. 2 illustrates linear interpolation of $\lambda(n)$ between $\lambda_{\min}(n)$ and $\lambda_{\max}(n)$ as a function of the current buffer fullness $B(n)$ and we obtain:

$$\lambda(n)=\frac{B_{\max}-B(n)}{B_{\max}-B_{\min}}\cdot\lambda_{\min}(n)+\frac{B(n)-B_{\min}}{B_{\max}-B_{\min}}\cdot\lambda_{\max}(n) \tag{7}$$

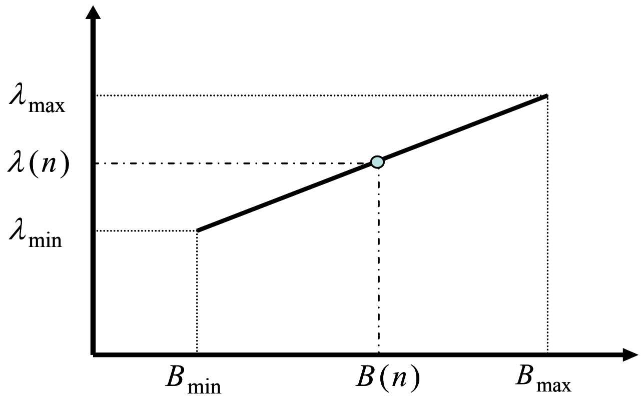

Figure 2: Linear interpolation of $\lambda(n)$ between $\lambda_{\min}(n)$ and $\lambda_{\max}(n)$ for the current buffer level $B(n)$.

3.2 Quadratic Interpolation (QI)

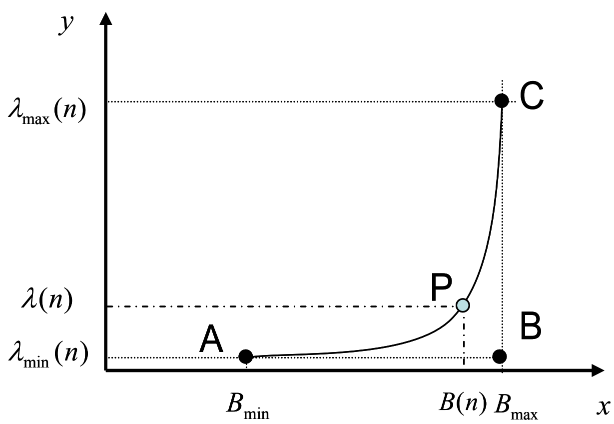

Figure 3: Quadratic interpolation of $\lambda(n)$ between $\lambda_{\min}(n)$ and $\lambda_{\max}(n)$ for the current buffer level $B(n)$.

Linear interpolation is the simplest way to interpolate $\lambda(n)$. An interpolation function that leads to more aggressive dropping if the buffer fullness comes close to $B_{\max}$ can be realized by quadratic interpolation of $\lambda(n)$, as shown in Fig. 3. With three control points A, B, and C, we can define a quadratic Bézier curve for $\lambda(n)$ with

$$A=(A_x,A_y)=(B_{\min},\lambda_{\min}(n))$$

$$B=(B_x,B_y)=(B_{\max},\lambda_{\min}(n))$$

$$C=(C_x,C_y)=(B_{\max},\lambda_{\max}(n))$$

$$P_x=(1-t)^2\cdot A_x+2t\cdot(1-t)\cdot B_x+t^2\cdot C_x \tag{8}$$

$$P_y=(1-t)^2\cdot A_y+2t\cdot(1-t)\cdot B_y+t^2\cdot C_y \tag{9}$$

The interpolated point $P=(P_x,P_y)$ moves on this curve from A to C by varying the parameter t from 0 to 1. For a given $B(n)$, we determine t and then $\lambda(n)=P_y$ from (8) and (9).

3.3 Computing $\lambda_{\min}(n)$ and $\lambda_{\max}(n)$

In order to determine $\lambda_{\min}(n)$ we evaluate (6) for every dropping pattern and select $\lambda_{\min}(n)$ such that the minimum of (6) is obtained for the dropping pattern where nothing is dropped in all K video streams. This means that

$$\begin{aligned} &J_{p_n}(n)\\ &=\sum_{k=1}^{K}\Delta D_{p_n}^{k}(n)-\lambda_{\min}(n)\sum_{k=1}^{K}\Delta R_{p_n}^{k}(n)\\ &\le\sum_{k=1}^{K}\Delta D_{p}^{k}(n)-\lambda_{\min}(n)\sum_{k=1}^{K}\Delta R_{p}^{k}(n)\\ &\quad \textit{for}\quad p=1\ldots P(n)\quad \textit{and}\quad p\neq p_n \end{aligned} \tag{10}$$

with p_n representing the pattern where no frame drop occurs in all video streams. As $J_{p_n}(n)$ equals zero, this leads to

$$\lambda_{\min}(n)\le\frac{\sum_{k=1}^{K}\Delta D_p^k(n)}{\sum_{k=1}^{K}\Delta R_p^k(n)} \tag{11}$$
$$\textit{for}\ \ p=1\ldots P(n)\quad \textit{and}\ \ p\neq p_n$$

and we pick $\lambda_{\min}(n)$ to be as big as possible while still satisfying all the inequalities in (11). The value for $\lambda_{\max}(n)$ is derived in a very similar fashion. For this, the minimization of (6) should now lead to the decision of dropping as many frames as possible (dropping pattern p_a), which leads to

$$\begin{aligned} &J_{p_a}(n)\\ &=\sum_{k=1}^{K}\Delta D_{p_a}^{k}(n)-\lambda_{\max}(n)\sum_{k=1}^{K}\Delta R_{p_a}^{k}(n)\\ &\le\sum_{k=1}^{K}\Delta D_{p}^{k}(n)-\lambda_{\max}(n)\sum_{k=1}^{K}\Delta R_{p}^{k}(n)\\ &\quad \text{for } p=1..P(n) \text{ and } p\neq p_a \end{aligned} \tag{12}$$

This results in

$$\lambda_{\max}(n) \geq \frac{\sum_{k=1}^{K}\left(\Delta D_{p_a}^k(n) - \Delta D_p^k(n)\right)}{\sum_{k=1}^{K}\left(\Delta R_{p_a}^k(n) - \Delta R_p^k(n)\right)} \quad (13)$$

for $p = 1..P(n)$ and $p \neq p_a$

and we pick $\lambda_{\max}(n)$ to be as small as possible while still satisfying all inequalities in (13).

3.4 Computational Complexity

The computational complexity of the dropping decision depends mainly on the number of possible dropping patterns. For each stream, we have at most four possible dropping decisions. So the total number of dropping patterns is at most 4^K, where K is the number of incoming streams. When the number of streams is very large, we can also group these streams and perform the dropping decisions independently for every group. For example, if we put always 4 streams in one group, then the complexity becomes a function of $K \cdot 4^{\frac{K}{4}-1}$. Please note that typically the number of dropping patterns is much smaller as previous dropping decisions reduce the current choices.

IV. SIMULATION RESULTS

We investigate how much improvement on the average reconstruction quality can be achieved by using the proposed rate-distortion optimized dropping strategy when compared to priority-based dropping. In our simulations we assume that four video streams that have been encoded with the emerging H.264 codec [3][4][5] arrive at an active network node and have to be sent out on the same outgoing link. Table 1 summarizes the main characteristics of the four videos.

Name	Length	Bit rate	PSNR(dB)
Foreman	299 frames	107kbps	36.28
Akiyo	279 frames	35kbps	38.94
Carphone	289 frames	106kbps	36.93
Grandmother	269 frames	42kbps	36.99

Table 1: Main characteristics of the four test videos.

The combined average rate of the four videos is 290 kbps. The actual rate at a certain time instant varies significantly because of the different frame types and the varying activity in the sequences. The GOP length of the four videos is $L^1 = 18$, $L^2 = 22$, $L^3 = 26$ and $L^4 = 24$ frames, respectively. The GOP structure for all videos is IBPBP… which corresponds to $B^k = 1$. The size of the outlink buffer is set to be 32KByte.

The simulations are performed for a video session length of 3000 frames. For this, the video sequences are continuously repeated. The simulation time is incremented in multiples of the frame period. This means that every frame period 4 new frames, one from each video, arrive at the network node for forwarding and a dropping decision is made.

Table 2 shows the overhead encountered when sending the side information along with the video bitstream. The overhead mainly depends on the GOP size when we assume the number of B frames in between P frames being fixed in the four streams.

	GOB size	D-M Entry /frame	R-V Entry /frame	Over-head/f bytes	Over-head kbps	Over-head %
Foreman	18	6	1	14	1.68	1.57
Akiyo	22	7	1	16	1.92	5.49
Carphone	26	8	1	18	2.16	2.04
Grandmother	24	7.5	1	17	2.04	4.86
Average						2.69

Table 2: Side information overhead of the four test videos.

The outlink buffer behaves in both cases (RD-optimized dropping and priority-based dropping) in exactly the same way. If a video frame is to be sent on the outgoing link, it is first placed in the output buffer. In case the buffer is too full to accommodate the frame, it is dropped. If more than one frame is sent to the outlink buffer at the same time, we assume that the packets are labeled according to their content (I-, B-, or P-frame). If the buffer cannot accommodate all frames it will always first drop the B-frames. If the buffer is still not empty enough for the remaining frames, P-frames are dropped next and in the same spirit, eventually the I-frames are dropped. This dropping mechanism corresponds to the priority-based dropping strategy used for comparison in this paper.

For the RD-optimized dropping strategy, the same outlink buffer is used but the active network node decides beforehand which frames to send to the buffer. Those frames that are decided to be dropped by minimization of (6) are not passed on to the buffer. Despite the optimization, it might still happen that more data is passed to the buffer than can be accommodated, which leads to additional loss of data. It is therefore possible that sometimes priority-based dropping also happens after an optimized dropping pattern has been determined.

In our experiments, we use the I-frame of the previous GOP as the replacement frame R in (3) during our dropping decision. When measuring reconstruction distortion at the receiver, however, we always use the most recent successfully decoded frame as a replacement for a dropped frame.

4.1. Performance comparison of RD-optimized frame dropping and priority-based dropping

Our RD-optimized dropping approach (RDOD) with linear and quadratic interpolation for the Lagrangian multiplier has been introduced in Section 3. In the simulation, we compare RDOD with priority-based random dropping (PRD). The working principle of PRD has been introduced above. Compared to traditional PRD, a performance improvement can be expected if the priority-based dropping is started before the buffer is 100% full. Dropping some B frames earlier when the buffer load exceeds a preset threshold B_1 and dropping some P frames when it exceeds threshold B_2 makes the prioritization of I versus B and P frames even stronger. We call this approach priority-based early random dropping (PRED). In our simulation, threshold B_1 and threshold B_2 are set to be 70% and 90% of $B_{\max}$. Here we set B_2 to be 90% because we want to make sure that I frames will not be dropped. The worst case is four I frames come in simultaneously and 10% is roughly the size of four I frames. However, this also depends on the size of the buffer. B_1 is set to be 70% because it has some space to hold some P frames before the buffer load comes to be 90% and also keep more possible B frames.

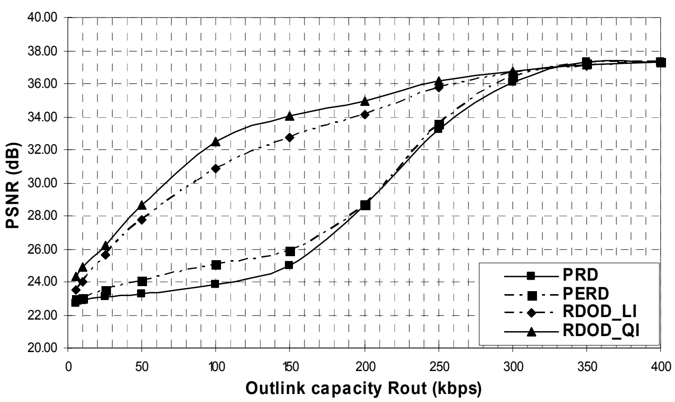

Figure 4: Video reconstruction quality vs. outlink rate.

Fig. 4 shows the improvements obtained by the RD-optimized video frame dropping concept proposed in this paper. The PSNR values are averaged over the 4 video sequences. When the outgoing bit-rate R_{out} is larger than the mean incoming rate (290 kbps), the RD-optimized dropping and the priority-based dropping perform similar. This is expected as in this case the buffer will very rarely overflow and only very few frames are lost in both cases. If, however, the outgoing rate is smaller than the total average rate of the 4 videos, it can be seen that the RD-optimized dropping leads to huge improvements in terms of reconstructed video quality. Quadratic interpolation (RDOD_QI) of $\lambda(n)$ leads to a better performance than linear interpolation (RDOD_LI). If we select the outgoing link rate to be $R_{out} = 150kbps$, we see an improvement of about 8 dB between the RDOD_QI and PERD.

	RDOD_QI (dB)	PRD (dB)	Gain (dB)
Foreman	32.8961	23.8787	9.0174
Akiyo	37.6096	33.4542	4.1554
Carphone	33.1991	25.6374	7.5617
Grandmother	35.8825	31.8644	4.0181
Average	34.8968	28.6937	6.2031

Table 3: Improvements observed for individual streams at $R_{out} = 200kbps$.

Table 3 shows the improvement obtained for individual streams. We can see that in our experiment we give more resources to the stream that has the highest motion between frames so that a smaller MSE caused by frame dropping can be achieved. Here `Foreman` has the strongest motion, although after optimization, it still gets the lowest PSNR. However it has the largest gain from the random dropping. Of course, this strategy is somehow unfair to the lower motion user. But this can also be adjusted by assigning different streams different weights of λ. In this case, the gain for `Akiyo` may increase while that of `Foreman` will decrease and the mean value of the gain will also be decreased.

4.2. Selecting $B_{\min}$

Only if the buffer level is larger than $B_{\min}$, the RD-optimized dropping strategy will start dropping video frames. Here we assume that $B_{\max}$ always corresponds to 100% fullness**.** The computation of $\lambda(n)$ in (8),(9) depends on $\lambda_{\min}(n)$ and $\lambda_{\max}(n)$ and therefore on the selection of $B_{\min}$. Fig. 5 shows the reconstructed video quality as a function of $B_{\min}$ for RDOD_QI. It can be observed that $B_{\min}$ has little influence on the reconstruction quality as long as we select it to be small enough. The outgoing rates for the simulation in Fig. 5 are 200 kbps and 250 kbps.

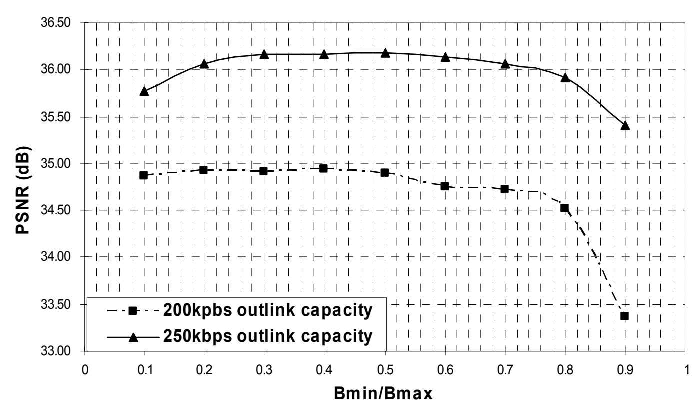

Figure 5: Average video reconstruction quality as a function of $B_{\min}/B_{\max}$ for two different outlink rates.

4.3 Low complexity RD-optimization

In Section 4.1 and 4.2, $\lambda(n)$ is re-computed every time new incoming data becomes available. When we use RDOP_QI, we observe that $\lambda(n)$ changes little over time as long as the buffer fullness $B(n)$ is smaller than $B_{\max}$. In order to reduce computational complexity, we consider calculating $\lambda(n)$ once and use this $\lambda(n)$ for the following m frames before we refresh the value of $\lambda(n)$. As shown in Fig. 6, only 0.3 dB quality decrease is observed when repeating $\lambda(n)$ for around 50 video frames.

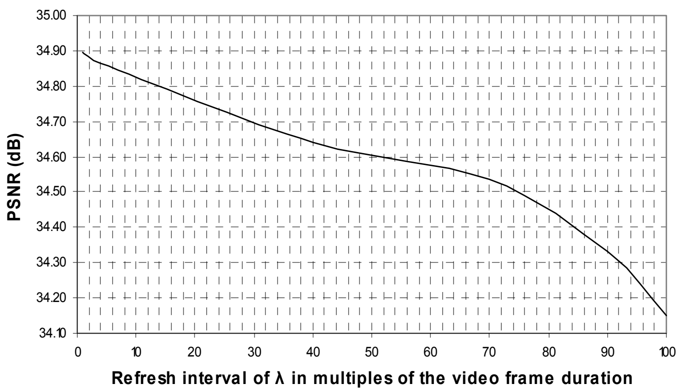

Figure 6: Average video reconstruction quality as a function of the refresh interval of λ.

V. CONCLUSIONS

We have presented an RD-optimized video frame dropping strategy that can be applied on active network nodes. The RD-optimization uses the rate vector and the distortion matrix to determine which frames should be dropped in case of heavy network load. The rate vector and the distortion matrix are sent as side information along with each GOP of the video. The only information extracted from the network node itself is the buffer fullness level. Significant quality improvements are reported when comparing our scheme to priority-based dropping. The presented concept could be in particular beneficial for mobile communication networks when applied, for example, in the base station to optimize the throughput of video streams for multiple cellular users.

VI. REFERENCES

[1] Ralph Keller, Sumi Choi, Dan Decasper, Marcel Dasen, George Fankhauser and Bernhard Plattner, “An Active Router Architecture for Multicast Video Distribution,” Proc. Infocom 2000, Tel Aviv, Israel, March 2000.

[2] D. Tennenhouse et al, “A Survey of Active Network Research,” IEEE Communications, January 1997.

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

[4] ITU-T Recommendation H.264, “Advanced video coding for generic audiovisual services”, May 2003.

[5] ISO/IEC International Standard 14496-10, MPEG-4 Part 10, Advanced Video Coding, 2003.

[6] G. Ravindra, N. Balakrishnan, K. R. Ramakrishnan, “Active Router Approach for Selective Packet Discard of Streamed MPEG Video under Low Bandwidth Conditions.” Proc. ICME 2000, New York, July 2000.

[7] Philip A. Chou and Zhourong Miao, “Rate-distortion optimized streaming of packetized media,” Tech. Rep. MSR-TR-2001-35, Microsoft Research, February 2001, (also submitted to IEEE Transactions on Multimedia).

[8] F. Zhai, R. Berry, T. N. Pappas, and A. K. Katsaggelos, “A rate-distortion optimized error control scheme for scalable video streaming over the Internet,” Proc. ICME 2003, Baltimore, MD, July 6-9, 2003.

[9] G.J. Sullivan and T. Wiegand, “Rate-Distortion Optimization for Video Compression,” IEEE Signal Processing Magazine, vol.15, no.6, pp.74-90, November 1998.

[10] I. Bouazizi, “Size-Distortion Optimized Proxy Caching for Robust Transmission of MPEG-4 Video,” In LNCS 2899, Proceedings International Workshop on Multimedia Interactive Protocols and Systems, Napoli, Italy, November 18-21, 2003.

[11] 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.