Modeling and Optimization of PET-Redundancy Assignment for MPEG Sequences

by Rainer Storn[1)]

TR-95-018

May 1995

Abstract

Priority Encoding Transmission (PET) is an encoding scheme which provides multiple levels of redundancy in order to protect the different contents of a data set according to their importance. The task of optimally assigning redundancies for the PET encoding scheme is investigated for the special case of MPEG-1 encoded video sequences. The prerequisites for this optimization problem and the way of proceeding for its solution are outlined and several suggestions for further improvements are given.

[1)]International Computer Science Institute, 1947 Center Street, Berkeley, CA 94704-1198, Suite 600, Fax: 510-643-7684. E-mail: storn@icsi.berkeley.edu. On leave from Siemens AG, ZFE T SN 2, Otto-Hahn-Ring 6, D-81739 Muenchen, Germany. Fax: 01149-636-44577,
E-mail: rainer.storn@zfe.siemens.de.

1. Introduction

The number of computer applications which make use of MPEG-1 encoded video sequences is continuously on the rise. At the same time many of these applications are intended for usage in a heterogeneous networking environment which exhibits links with different bandwidths and workstations with varying computational power. A major goal of priority encoding transmission (PET) is to provide graceful degradation for MPEG-1 encoded video streams in order to allow users with different HW-equipment to share the same applications [1]. To this end PET protects the different contents of an MPEG-1 video according to their importance via a multilevel redundancy scheme and information spreading. The intention is to increase the likelihood that the more important parts of the information can still be recovered if the video stream is corrupted by packet losses. These packet losses can be caused by different events like network congestion, fading in satellite transmission, insufficient computational power of the receiving workstations and the like. Coding the video streams in such a way that a potential degradation is perceived as graceful is a challenging task. First it must be clear how the different quality reductions in a video sequence are perceived by the human visual system. These reductions affect basically colour, spatial and temporal behaviour of the video as well as overall precision of the pixels. A thorough understanding of the MPEG-1 as well as the PET encoding process is then necessary to analyze what kind of changes lead to which effect, assuming that a certain packet loss behaviour is given. An optimized encoding scheme should yield the best video quality for a certain pattern of disturbance.

2. MPEG-1 Compression

MPEG (Moving Picture Experts Group) is a standardization group which has defined a widely accepted compression standard for video sequences in the 1.5Mbps range. This standard, the development of which was primarily driven by the rapidly expanding CD-ROM-Video market, is often called MPEG but should be called more accurately MPEG-1 as there is a second proposal, MPEG-2 [2], which addresses digital television and is still under scrutiny.

An MPEG-1-video is represented by a layered bistream as being described in [3]. The top layer is called a sequence which is encapsulated by the sequence delimiters header and trailer. The sequence header contains information such as picture size, picture rate and bit rate. The trailer consists of a 32bit end code.

The header is followed by any number of GOPs (Group of Pictures). A GOP provides random access, since it is the smallest coding unit which, under certain circumstances, can be decoded independently.

A GOP consists of a GOP header being followed by different types of frames. MPEG-1 employs three frame types: intraframe frames (I-frames), predictive frames (P-frames) and bi-directionally interpolated frames (B-frames) as indicated in Fig.1. I-frames, which are self contained and can be decoded independently, are basically encoded like JPEG pictures as illustrated in Fig. 2. JPEG stands for Joint Photographic Experts Group and also defines an international digital image compression standard for continuous-tone (multilevel) still images [4]. In every GOP there is one I-frame which immediately follows the GOP header if the GOP is viewed in bitstream order (further details concerning bitstream and display order are described in [2]). Due to the perceptive properties of the human visual system the luminance component is sampled with more spatial accuracy than the chrominance components. Each chrominance component, chrominance red and chrominance blue, has one-half the vertical and horizontal resolution of the luminance component. A so-called macroblock in MPEG-1 consists of 4 luminance blocks and just 2 chrominance blocks as shown in Fig. 3.

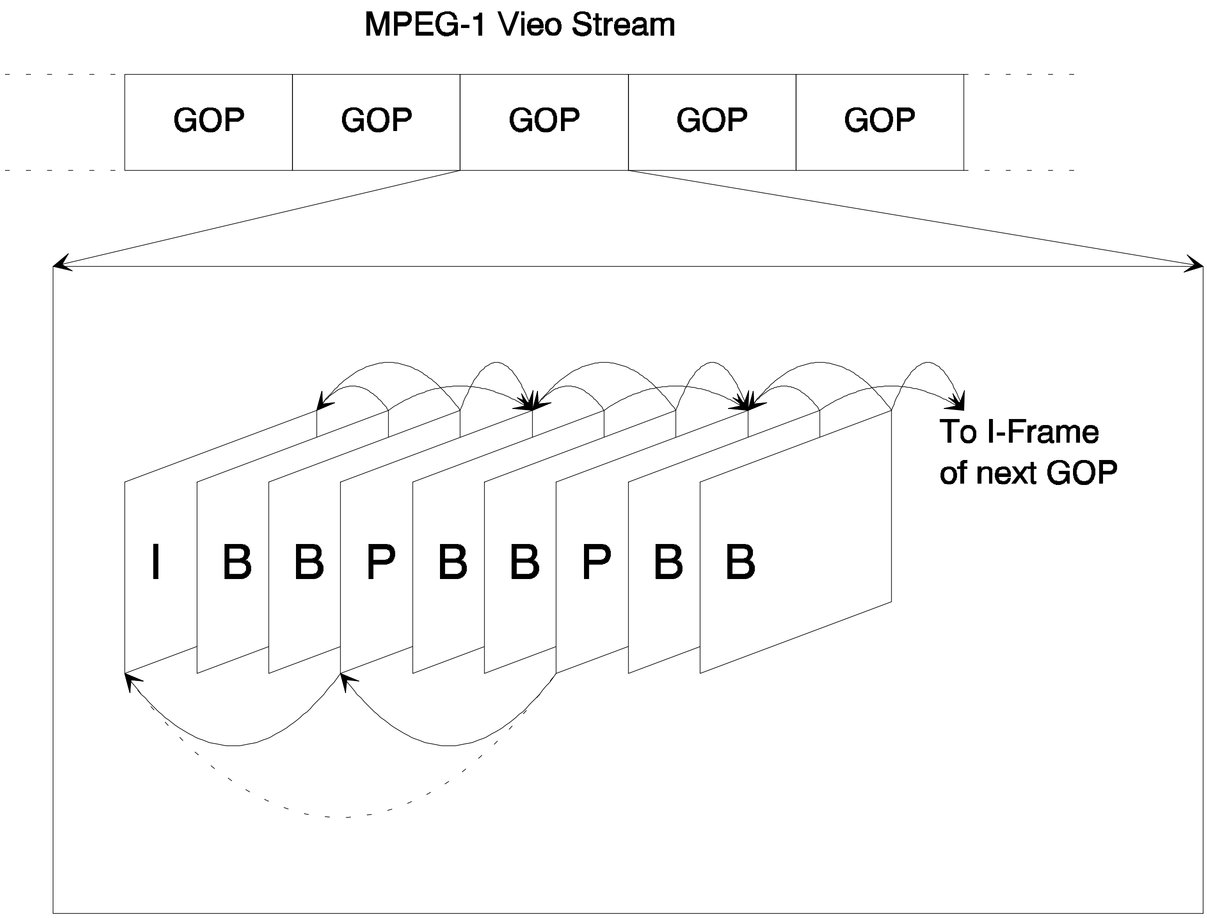

Fig.1: Example of Interframe coding of an MPEG-1 video stream without the GOP-flag being set.

As indicated in Fig. 1 P-frames need a past I- or P-frame for decoding making use of motion-compensated prediction. Motion-compensated prediction assumes that "locally" the current picture can be modeled as a translation of the picture at some previous time. Hence, in order to code a certain block of a P-frame, the previous I- or P-frame is used as a reference and searched for a block which exhibits the least difference to the block to be encoded (Fig. 4). Once a good match has been found, the displacement and an error term are used to encode the block under scrutiny. The comparison search is fairly computation expensive which is why the search is restricted to the luminance component of the block. After the search, however, the actual coding is done on luminance as well as chrominance.

B-Frames are encoded with motion vectors from a past (I- or P-) and a future (I- or P-) frame via interpolative techniques. B-frames are not used for prediction of other frames so the errors in them do not propagate, as opposed to the errors in I- or P-frames. This makes the B-frames the least critical of the three frame types. Errors in I- and P-frames propagate until the end of the GOP if the GOP has its "closed GOP"-flag set [2]. If the flag is not set the error in a P-frame can even propagate into the next GOP, and an error in the I-frame might affect B-frames in the previous GOP.

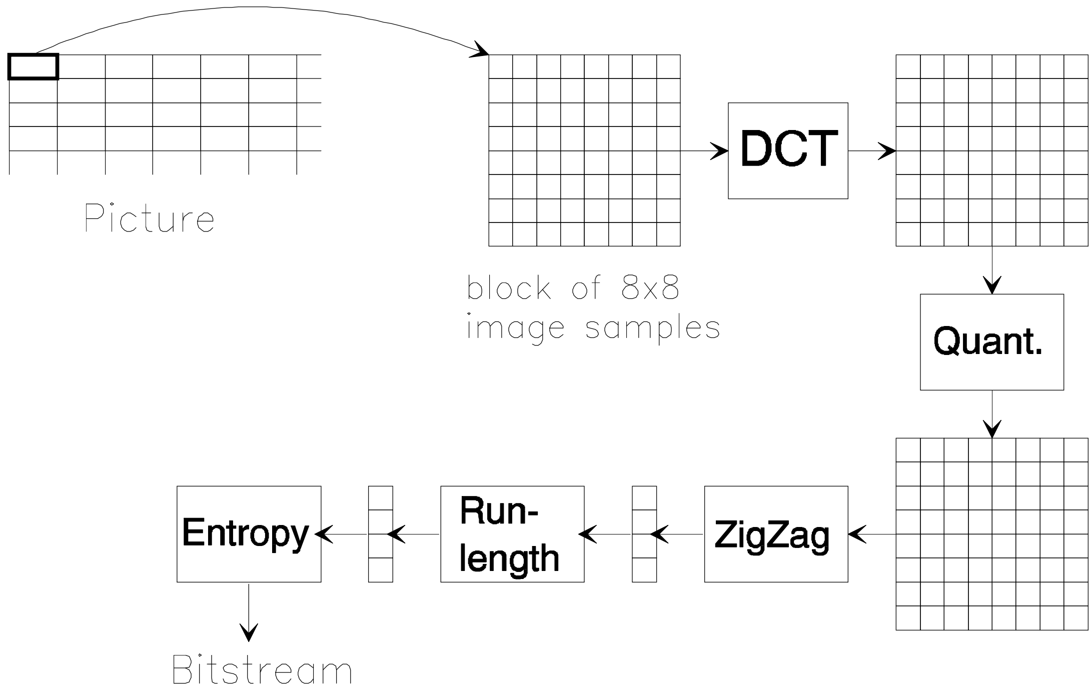

Fig.2: Block Transfer Encoding, the basis of JPEG and MPEG encoding.

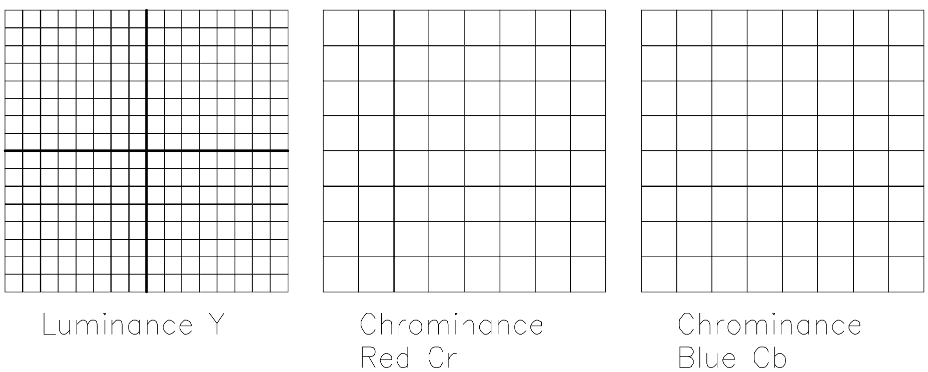

Fig. 3: A Macroblock in MPEG-1 consists of four luminance blocks and two chrominance blocks.

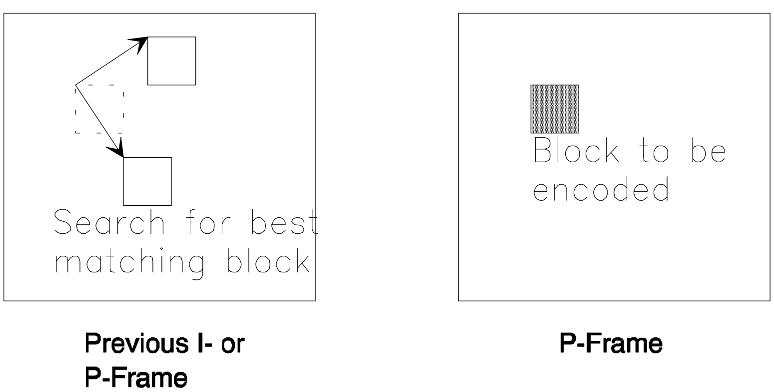

Fig. 4: Motion compensation for a P-Frame.

Another point worth mentioning concerns the size of the frames. There is no such thing as a fixed size for all three frame types. The I-frame size can vary in the different GOPs of a video sequence depending on the scene content. The sizes of P- and B-frames can vary even within a GOP. In addition the MPEG-1 standard allows the pattern of I-, P- and B-frames to vary dynamically between different GOPs although this is usually not done [5].

In order to simplify our further discussion, it will be assumed that

1) the GOP pattern shall be constant throughout a video stream.
2) P-frame errors do not propagate outside a GOP.
3) the sizes of I-, P- and B-frames shall be constant within a certain GOP and shall take on their mean value with respect to the entire GOP.
4) the influence of the GOP-header will be neglected.

Two examples of GOPs are given in Table 1:

Video	Red's Nightmare	Ice Hockey Sequence
GOP-pattern (display order)	I(9xB)P(9xB)P(9xB)	I(2xB)P(2xB)P(2xB)P(2xB) P(2xB)
mean I-frame size / kbyte	12.5	12.543
mean P-frame size / kbyte	8.1	10.411
mean B-frame size / kbyte	2.25	4.593
N_i=I-frame size*#frames	12.5	12.54
N_p=P-frame size*#frames	16.2	41.64
N_b=B-frame size*#frames	60.8	45.93
N_{gop}=size of GOP / kbyte	89.5	100.11

Table 1: Examples of frame sizes for two video sequences.

3. Priority Encoding Transmission (PET)

From chapter 2 it has become clear that due to the different error propogation properties I-, P- and B-frames differ in their importance concerning the information content of a GOP. The main idea in PET is to assign a certain amount of redundancy to a GOP with the redundancy being unevenly distributed among the different frame types, according to their importance [3], [8]. Most probably it is advantageous not to restrict the redundancy assignment just to the level of frame types if graceful degradation is the overall goal. These issues, however, will be discussed in a later chapter.The optimum partitioning of the redundancy among the three frame types shall be the main concern at first. At present PET is implemented in two versions with version 1 being based on the encoding of information via polynomials over Galois Fields [3], [8] and version 2 being based on encoding of information via Cauchy matrices [8]. Only version 2 will be regarded as this method allows faster encoding and decoding speeds and yields improved information recovery properties.

3.1 A simplified view of PET

Only those issues of the PET scheme will be described that are crucial for the decision process of optimally assigning priorities. This simplified view neglects several details of PET but still is appropriate for describing the problem of priority assignment. The basics of PET can be explained with the aid of Fig. 5. It can be seen that the GOP, which constitutes the message to be encoded, is encoded into n packets of a certain size such that the total size of all n packets amounts to N kBytes. The mapping of the GOP onto the n packets is done in such a way that information from every frame is contained in each of the packets. As a consequence the information is spread among the n packets which renders improved robustness in the presence of bursty errors which are common in today' s networking environment. The idea of information spreading has a long tradition and has been used extensively, e.g. in spread spectrum modulation. The second idea in PET is to provide error correcting properties on a multilevel basis. I.e., The most important data, the I-frame is endowed with relatively more redundancy information than the less important P- and B-frames. A nice property of PET is the fact that no decoding is required at the receiver if the cleartext information arrives undisturbed. This feature allows for fast processing in case of error free environments. It shall be noted as an aside that the PET version which uses Galois Fields doesn' t posess this property as the GOP is totally encrypted, and hence decoding is always required.

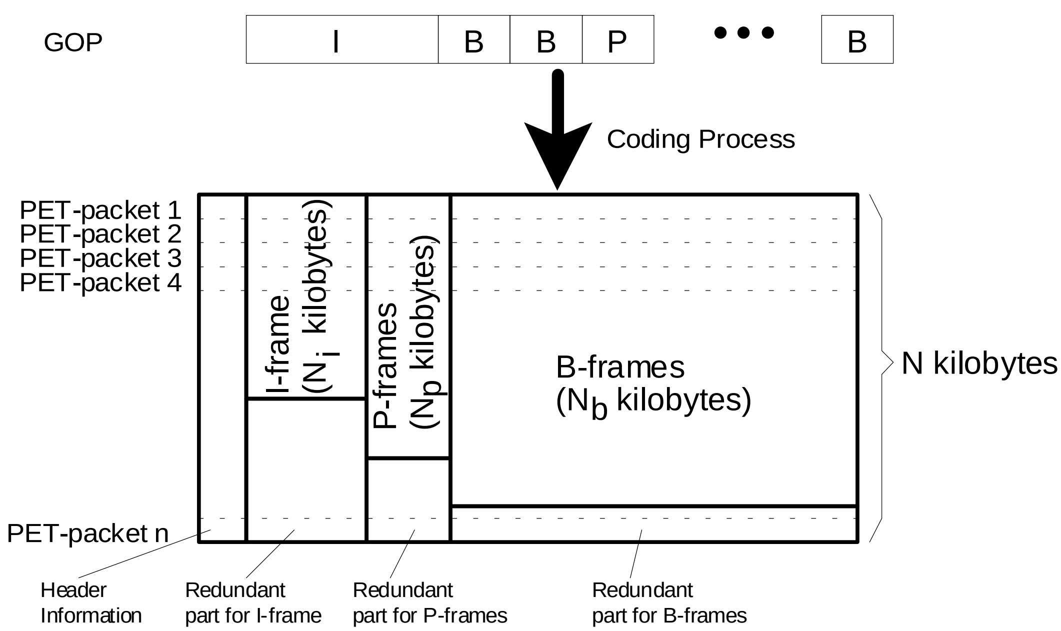

Fig. 5:Illustration of the coding process being effective in PET.

In case of errors the amount of redundancy being assigned to the different frame types decides whether the frames of the specific type can be recovered or not. If enough error free packets arrive, all of the frames belonging to a certain frame type can be recovered. If this threshold is not reached, no frame recovery is possible via decoding. Nevertheless, some cleartext information might have gotten through so that there is still a chance that some usable information has arrived, even though the recovery mechanism doesn' t have enough packets to recover the cleartext information. For the sake of simplicity it will be assumed, however, that the packet stream will always be decoded and no access to single cleartext information is possible. This leads to an "all or nothing" strategy where either all frames of a certain type or no frame of this type can be reconstructed. In a later chapter this assumption will be dropped.

The simplified PET scheme will be analyzed more deeply in the following. Consider the following definitions:

N_i = size in kBytes of an I-frame times the number of I-frames within a GOP.

N_p = size in kBytes of a P-frame times the number of P-frames within a GOP.

N_b = size in kBytes of a B-frame times the number of B-frames within a GOP.

N = size in kBytes of all n PET-packets

n_i = number of packets required to reconstruct all I-frames out of the n packets.

n_p = number of packets required to reconstruct all P-frames out of the n packets.

n_b = number of packets required to reconstruct all B-frames out of the n packets.

and $x_k = \frac{n_k}{n}$, k=i,p,b, where x_k represents the fraction of packets that are necessary to recover all frames of type k. The x_k will also be referred to as the PET parameters which shall be chosen optimally to get the best possible video quality.

With the above definitions it can be shown [8] that the following equations hold:

$$N = \frac{N_i}{x_i} + \frac{N_p}{x_p} + \frac{N_b}{x_b} \qquad (1)$$

and

$$N = (N_i + N_p + N_b) \cdot (1 + r) \qquad (2)$$

with r denoting the overall redundancy factor that is spent in the encoding process. Note that the x_k in (1) are dependent on r.

The PET parameters x_k are restricted to values from the range [0,1] and represent the unknowns while all other variables in (1) are supposed to be known. If we recast (1) according to

$$0 \le x_b = \frac{N_b}{N - \frac{N_i}{x_i} - \frac{N_p}{x_p}} \le 1 \qquad (3)$$

we can clearly see the condition which will make the optimum choice of the x_k a constrained optimization problem. If we consider the left part of (3) we obtain the boundary function

$$x_p = \frac{N_p}{N - N_b} + \frac{N_i \cdot N_p}{(N - N_b) \cdot ((N - N_b) \cdot x_i - N_i)} \qquad (4)$$

which is shown in Fig. 6 for the video sequence "Red' s Nightmare" from Table 1 and r=0.3.

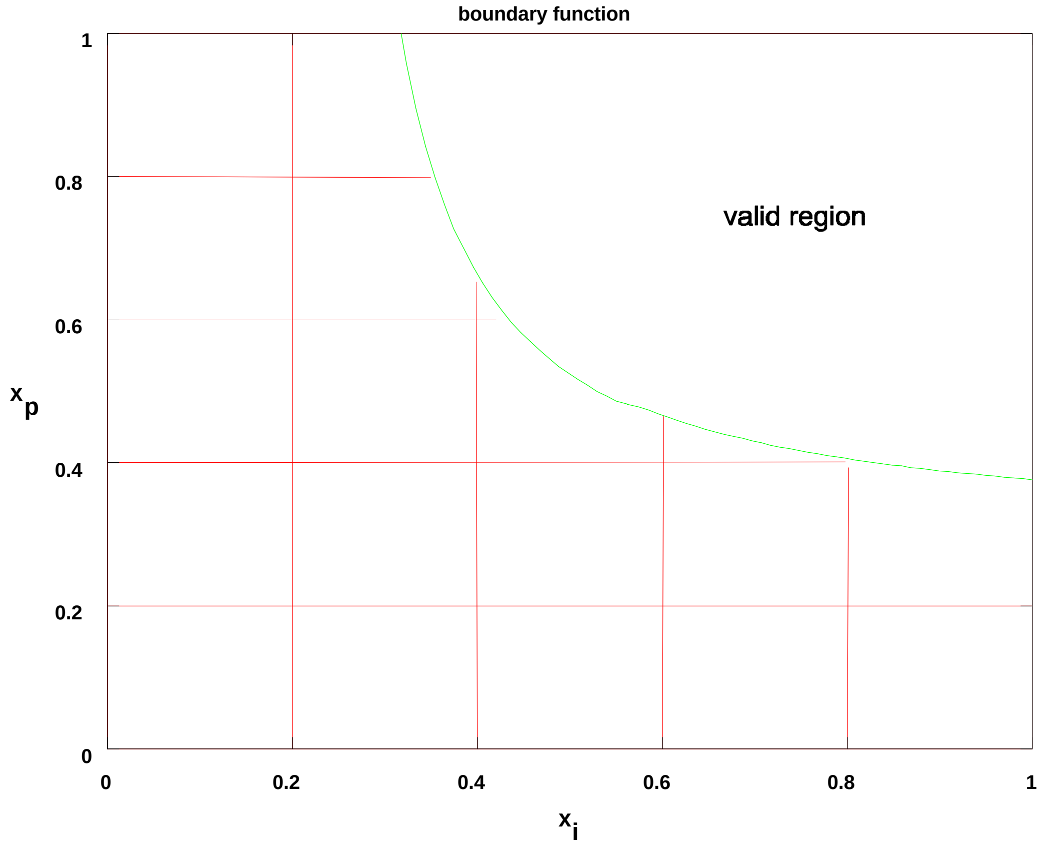

Fig. 6: Constraint for x_p and x_i for the video "Red' s Nightmare" and 30% redundancy.

3.2 Packet Loss Probabilities

In addition to the redundancy which is going to be spent for the encoding process the probability of loosing packets also has influence on the choice of the parameters x_k, k=i,p,b. This probability is, among other things, dependent on the type of network, the load of the network, the packet size, environmental conditions (if e.g. satellite links make up parts of the network), the burstiness of the traffic and correlations. At ICSI measurements over a moderately loaded Internet using 130,000 packets of size 2kByte yielded the packet loss probability distribution shown in figure 7. Note that the probability distribution in Fig. 7 is restricted to the case where the GOP is encoded into 100 packets with length 2kByte. If one of the two parameters, number or size of packets, changes, the probabilities will change. As different videos are generally encoded in a different way, it ensues that it is virtually impossible to undertake the optimization process with just using one probability density function (pdf), even if the network load should remain unaltered.There are, however, two ways to reuse the packet loss measurements if the packet size is kept constant:

1) The total packet stream of the measurement (in our case 130,000 packets) can be used to compute packet loss statistics for different batch sizes.

2) If the packet stream and the measurement results are no longer available, even the probability density function (pdf) can be reused to a certain extent to approximate pdfs for different batch sizes (but identical packet length).

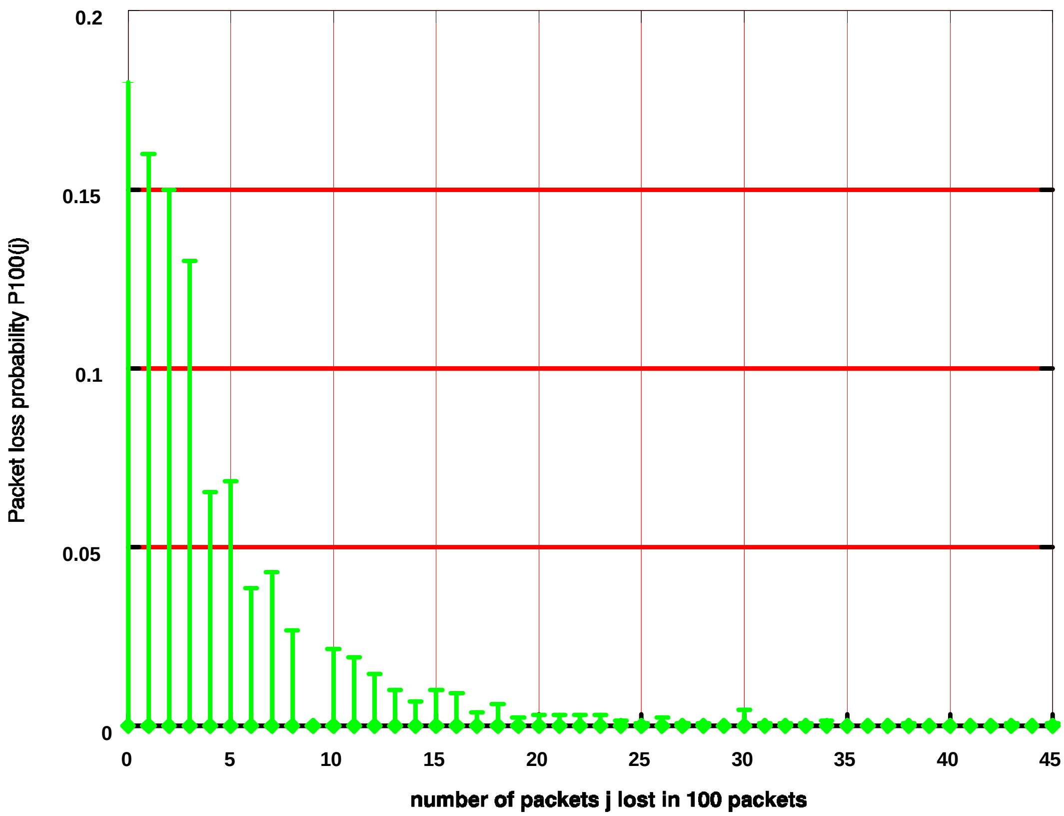

Fig. 7: Packet loss probability for 2KByte-sized packets over a moderately loaded Internet.

3.3 Derivation of a pdf for different batch sizes from the original pdf

Sometimes the only source of information concerning the packet loss behaviour is the pdf for a certain batch size. The reasons for that could be that the original test sequences and measurement results have been deleted with the pdf being the only result left. Yet there are still some possibilities to reuse this pdf for other batch sizes than the original one. Let' s regard a batch of n packets which has supposedly been used for the computation of the original pdf. Let' s further divide this batch into m equal partitions as indicated in Fig. 8.

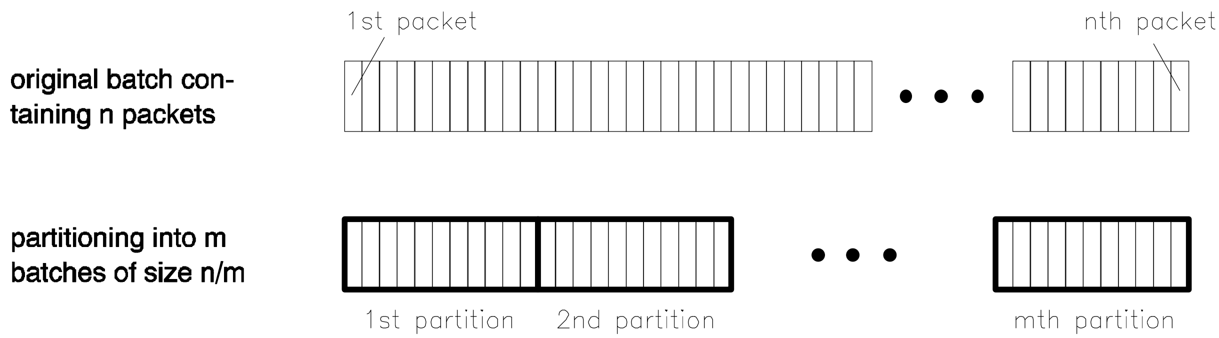

Fig. 8 Partitioning of an n-sized batch into m n/m-sized batches.

Consider the simple case where just one packet in the original n-sized batch shall be destroyed. The probability that this error hits a certain batch out of the m n/m-sized batches is 1/m. This holds true for each of the m batches, so it suffices to regard just one of them, let's say the first batch.

In general we might assume that k packet losses affect the n-sized batch. The number of possibilities to distribute these k losses, k<=n/m, among the m n/m-sized batches is $\binom{m+k-1}{k}$.

Consider the case where just one packet loss occurs in the first n/m-sized batch. Then there are k-1 losses left which can be distributed among the remaining (m-1) batches in $\binom{(m-1)+(k-1)-1}{(k-1)}$ variations. Hence the probability of getting just one packet loss in an n/m-sized batch although k losses occur in the n-sized batch is clearly $\frac{\binom{(m-1)+(k-1)-1}{(k-1)}}{\binom{m+k-1}{k}}$.

We can generalize these findings to get the probability of i losses in an n/m-sized batch while k losses occur in the n-sized batch. As a result we get $\frac{\binom{(m-1)+(k-i)-1}{(k-i)}}{\binom{m+k-1}{k}}$.

Finally we can compute the probability of i losses in an n/m-sized batch $p_{\frac{n}{m},i}$ via

$$p_{\frac{n}{m},i} = \sum_{k=i}^{k_{max}} \frac{\binom{(m-1)+(k-i)-1}{(k-i)}}{\binom{m+k-1}{k}} \cdot p_{n,k} \qquad (5)$$

with $p_{n,k}$ being the probability of having k losses in an n-sized batch. Of course (5) is only valid if the restrictions i<= k_{max}<= n/m and n/m integer apply. Even then (5) is only an approximation as all packet loss probabilities for more than n/m losses are neglected and it is assumed that all error patterns are equally probable.

4. The optimization task

Let's assume that the pdf for the packet loss belonging to a particular video stream is available. Also recall that x_k is the fraction out of n packets required to reconstruct all frames of type k. If we apply the redundancy r to the video stream in a uniform manner we see from (1) and (2) that there is only one fractional value $x = x_i = x_p = x_b = \frac{1}{1+r}$ which determines the recovery threshold. If x is the required fraction of packets to be received, 1-x is the fraction which is allowed to be lost. The probability for loosing the fraction 1-x can be determined by summing all packet loss probabilities from 1-x to 1. As illustrated in Fig 9a, all information will be recoverable if not more than $n \cdot (1-x) = \frac{n \cdot r}{1+r}$ packets are lost. A packet loss less than this number will result in complete recovery and hence good video quality. If more than $\frac{n \cdot r}{1+r}$ packets are lost, no recovery is possible at all and the video quality will, of course, be unacceptable. A multilevel PET should render results which are qualitatively described by Fig. 9b. E.g., if more than $n \cdot (1-x_b)$ packets are lost, the B-frames are not recoverable any more, but P- and I-frames can still be recovered if the number of lost packets in a batch doesn't exceed $n \cdot (1-x_p)$ or $n \cdot (1-x_i)$ respectively. In this case the video quality is labeled fair. The multilevel PET scheme generates a transition band between good and bad video quality with less chance of a bad video. The price to pay is that the threshold where the good video quality starts to be degraded is lower compared to the single-level case. In order to obtain the most satisfying results one has to tackle the task of optimizing the parameters x_i, x_p and x_b.

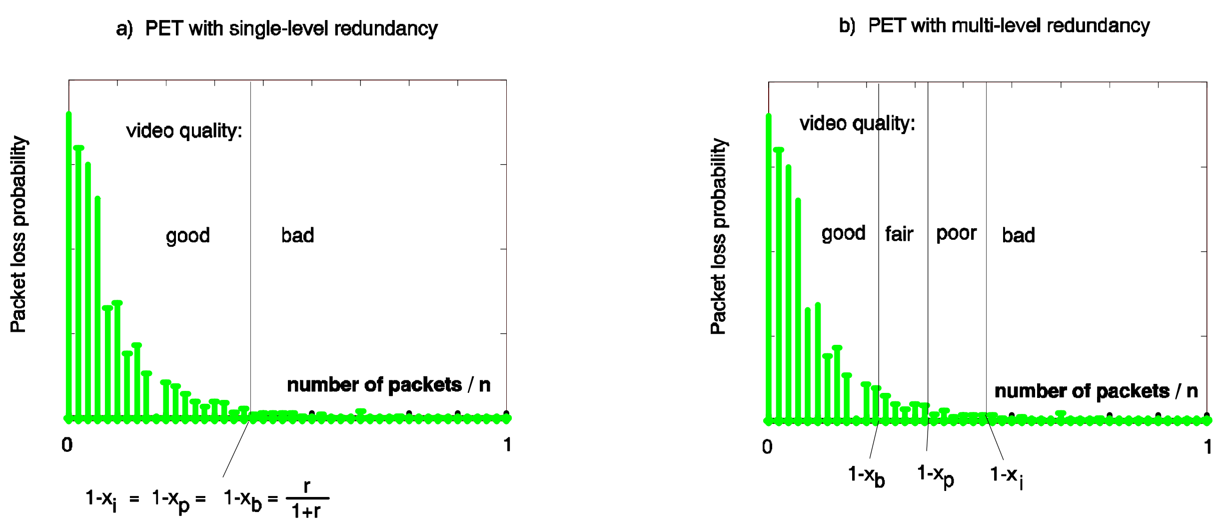

Fig. 9: Qualitative effect of a multi-level PET-scheme.

Before choosing the parameters x_k, k=i,p,b, the objective of the optimization process has to be defined. Most optimization procedures require the definition of an objective function the value of which has to be minimized.

Let p_{n,n_k} be the probability of loosing n_k or more packets assuming that n packets have been used to encode a GOP. If

$$n_k = n \cdot (1-x_k) \tag{6}$$

then p_{n,n_k} describes the probability of loosing all frames of type k. Recall again that it had been assumed that the PET scheme can either recover all frames of a certain type or loose them all depending on the amount of error-free packets that are available.

The choice of the objective function is crucial to the entire optimization process. The objective function must be able to model the degradation behaviour of a video stream in such a way that the minimization of the objective function renders the most convenient degradation of the video. In a straightforward manner we choose the objective function to be

$$f = p_{n,n_i} \cdot k_i + p_{n,n_p} \cdot k_p + p_{n,n_b} \cdot k_b, \tag{7}$$

a function which is based on technical rather than perceptive considerations. The constants k_i, k_p and k_b are weights which have to be chosen according to some criteria which will be discussed below. The objective function f is apt to represent a weighted expectation of the lost information in a GOP due to packet losses. The term p_{n,n_i} accounts for the probability of loosing all I-frames, the terms p_{n,n_p} and p_{n,n_b} do the same for the P- and B-frames respectively. We implicitly assume here that the video quality gets worse in proportion to the amount of lost information.

4.1 Choice of the weighting factors

The choice of the weighting factors k_i, k_p and k_b can be done in several ways.

CASE 1: Weighting factors proportional to number of lost frames

The simplest approach is to choose the weighting factors according to the number of lost frames. An example will illustrate this approach: The GOP of the video "Red's Nightmare" has the pattern I(9xB)P(9xB)P(9xB) and exhibits a total of 30 frames. If the I-frame cannot be regenerated the entire GOP cannot be decoded. In addition, the last 9 B-frames of the previous GOP are assumed to be affected also. If the P-frames are lost the entire GOP except the I-frame cannot be reconstructed which yields a total of 29 lost frames. If all B-frames are lost, 27 frames are destroyed. Hence

$$k_i = 39,\ k_p = 29,\ k_b = 27 \tag{8}$$

will be an appropriate choice for the weighting factors. The resulting objective function f in case of an overall redundancy of r=10% is shown in Fig. 10. Note also that the objective function in Fig. 10 has been computed by employing the pdf of Fig. 7 and hence assumes that Red's Nightmare is encoded in 100 2kByte packets. The value of f in case of the forbidden region for the parameter set x_p, x_i has been set to 100 for convenience.

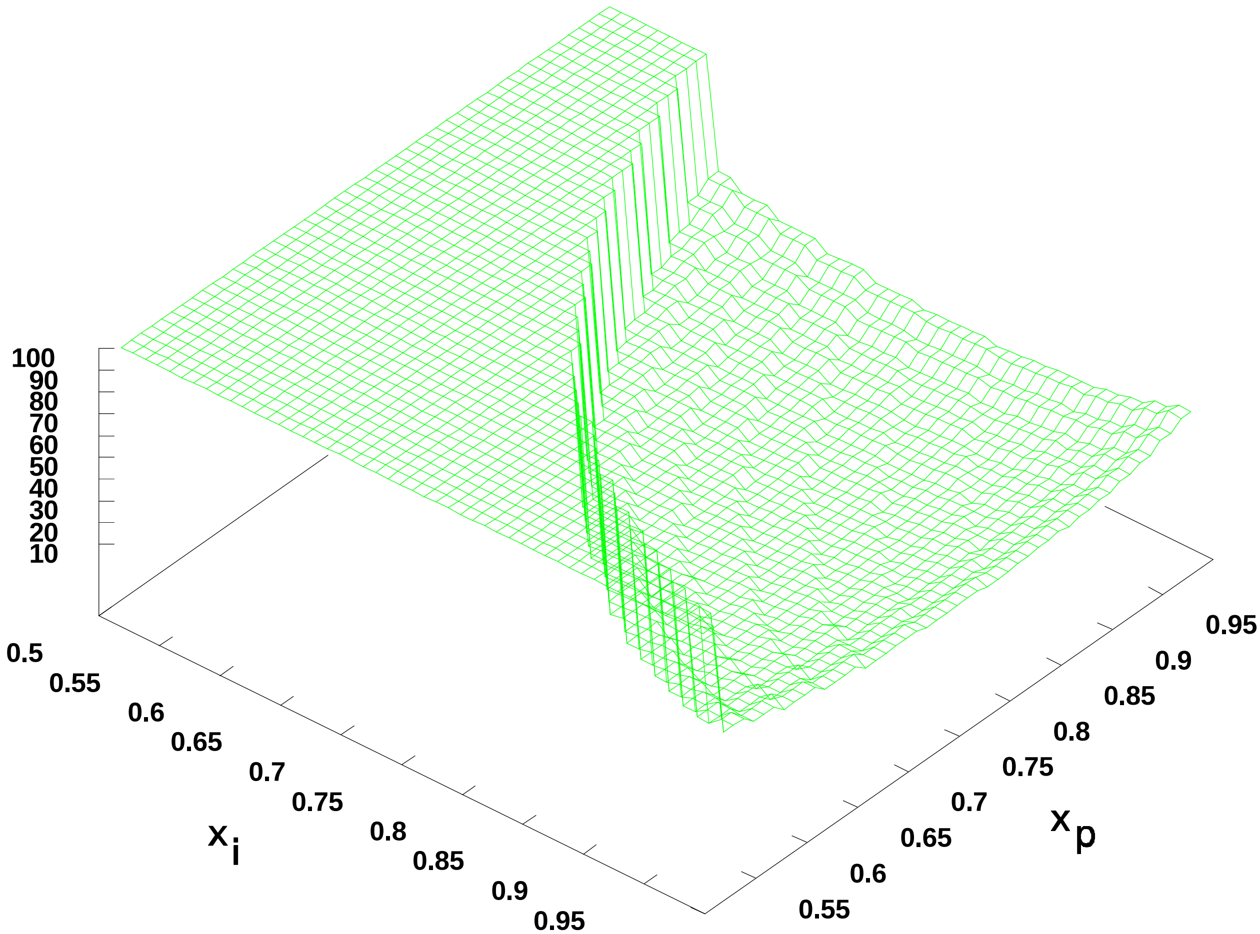

Fig. 10: Objective function for "Red's Nightmare", CASE1, and an overall redundancy of r=10%.

CASE 2: Weighting factor proportional to the number of lost bytes

This choice is similiar to CASE 1 but consideres the fact that the different frame types have different size. Hence CASE 2 provides a more accurate incarnation of the "video degradation proportional to lost information" paradigma. If we take kbytes as the dimension, we obtain

$k_i = 12.5 \cdot 1 + 8.1 \cdot 2 + 2.25 \cdot 39 = 116.45$ (9a)

$k_p = 8.1 \cdot 2 + 2.25 \cdot 27 = 76.95$ (9b)

$k_b = 2.25 \cdot 27 = 60.75$ (9c)

for the "Red's Nightmare" example. An example plot of the corresponding objective function for r=10% is depicted in Fig. 11.

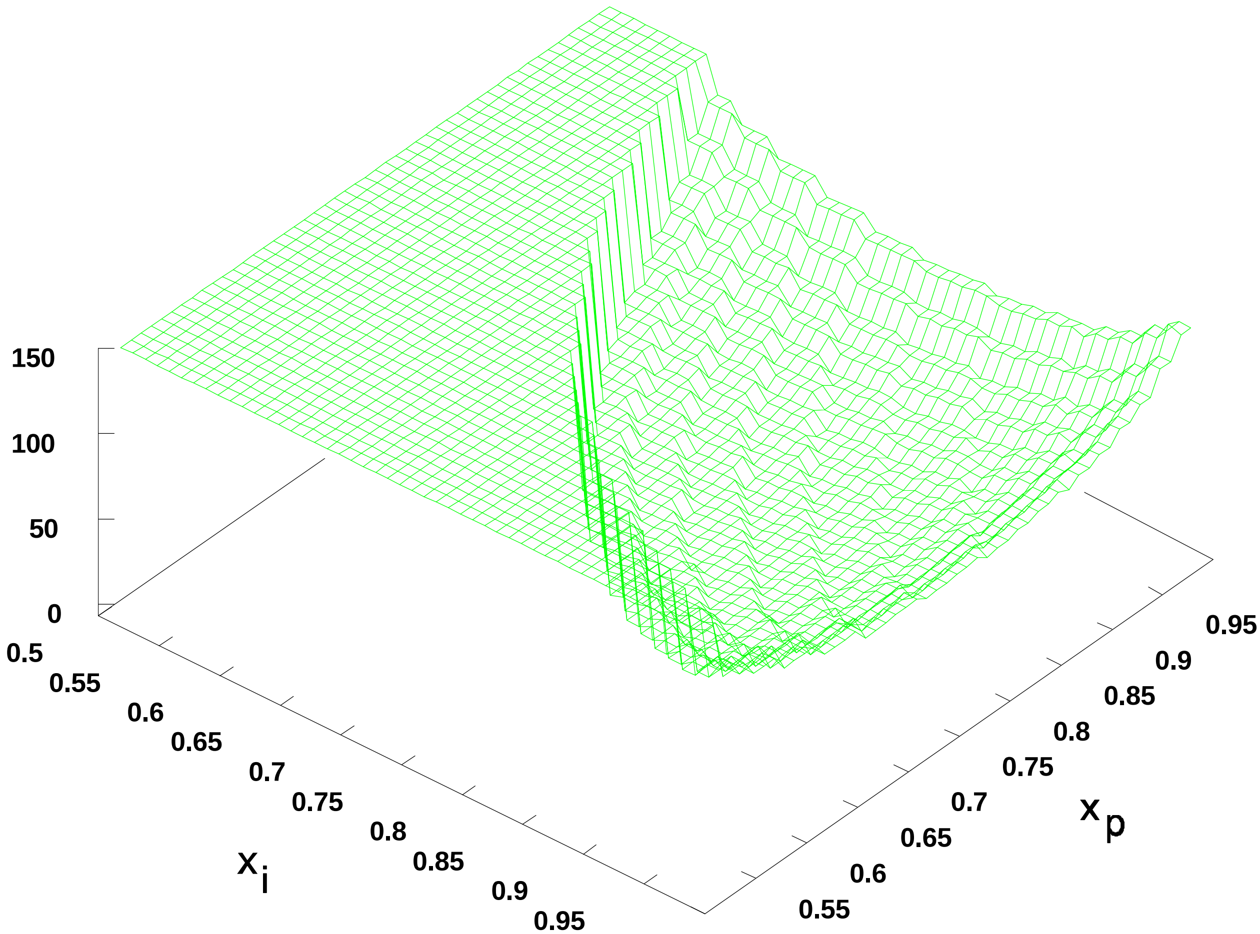

Fig. 11: Objective function for "Red's Nightmare", CASE2, and an overall redundancy of r=10%.

CASE 3: Weighting factor proportional to granularity

It is well known that the human being is very sensitive to disruptions of rhythms. Hence another choice of the weighting factors has been contrived which assumes that a frame is the more important the coarser its granularity. The I-frame in "Red's Nightmare", for example, occurs every 30th frame and hence $k_i = 30$ is chosen. The P-frames occur roughly every 9th to 10th frame which renders the choice $k_p = 10$. The B-frames exhibit a granularity of approximately 1 which yields $k_b = 1$. The corresponding objective function with r=10% is shown in fig. 12.

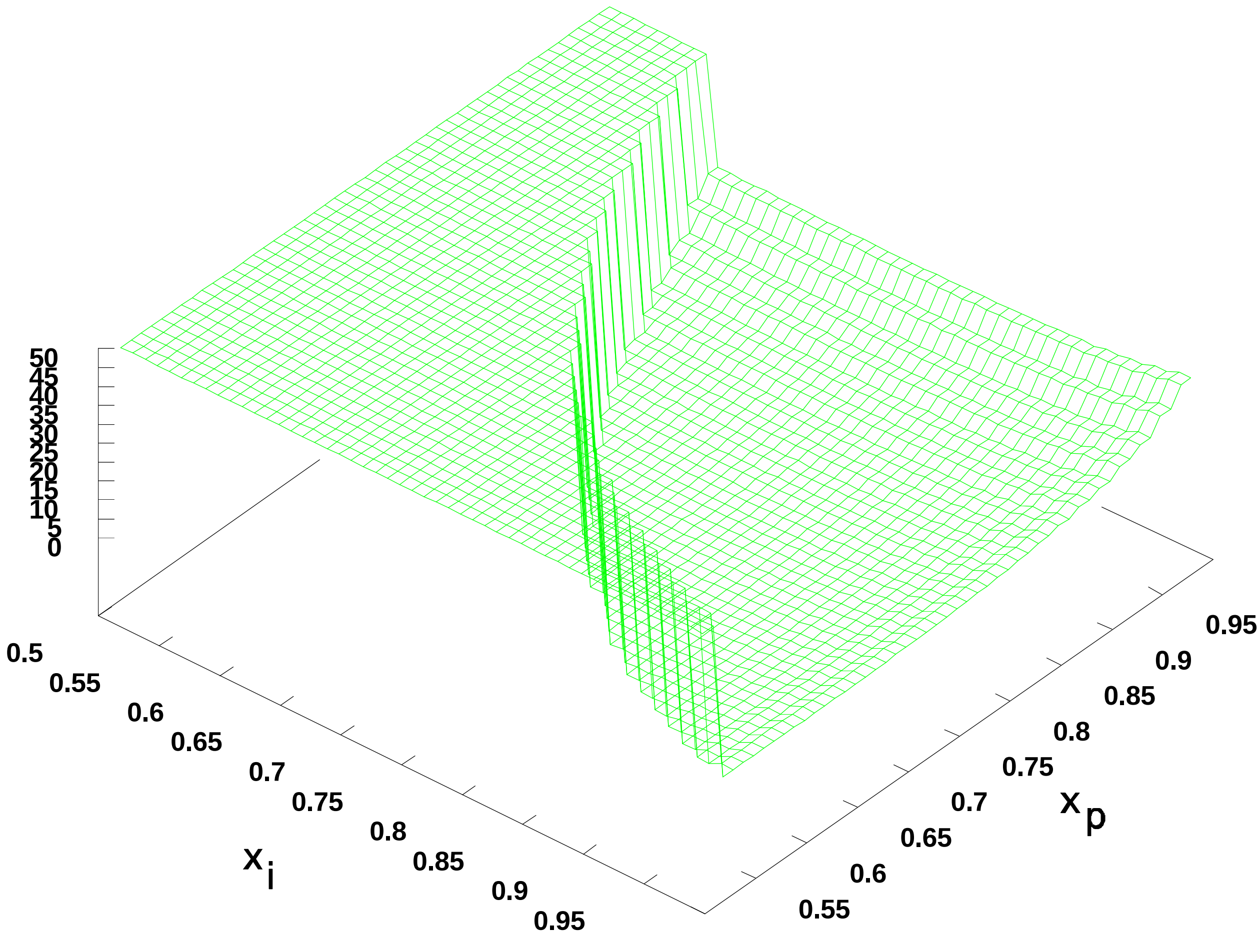

Fig. 12: Objective function for "Red's Nightmare", CASE3, and an overall redundancy of r=10%.

CASE 4: Weighting factors for evenly distributed redundancy

Videostreams with prioritized redundancy call for a comparison with streams that exhibit unprioritized redundancy. Although the fractions x_i, x_p and x_b can be computed directly from (1) and (2) via

$$x = x_i = x_p = x_b = \frac{1}{1+r} \ , \tag{10}$$

one might be curious which choice of k_i, k_p and k_b corresponds to (10).

It turns out that the weights must be chosen proportional to the number of bytes which make up the collection of all frames of a certain kind in a GOP. For the example "Red's Nightmare" we would obtain

$$k_i = 12.5, k_p = 16.2, k_b = 60.8. \tag{11}$$

This renders e.g. $k_p > k_i$ as the P-frames contribute more data mass to the GOP than the I-frame. The above choice for k_i, k_p and k_b results in every bit of the GOP getting the same weight.

Fig. 13 shows an illustration of the corresponding objective function with r = 10%.

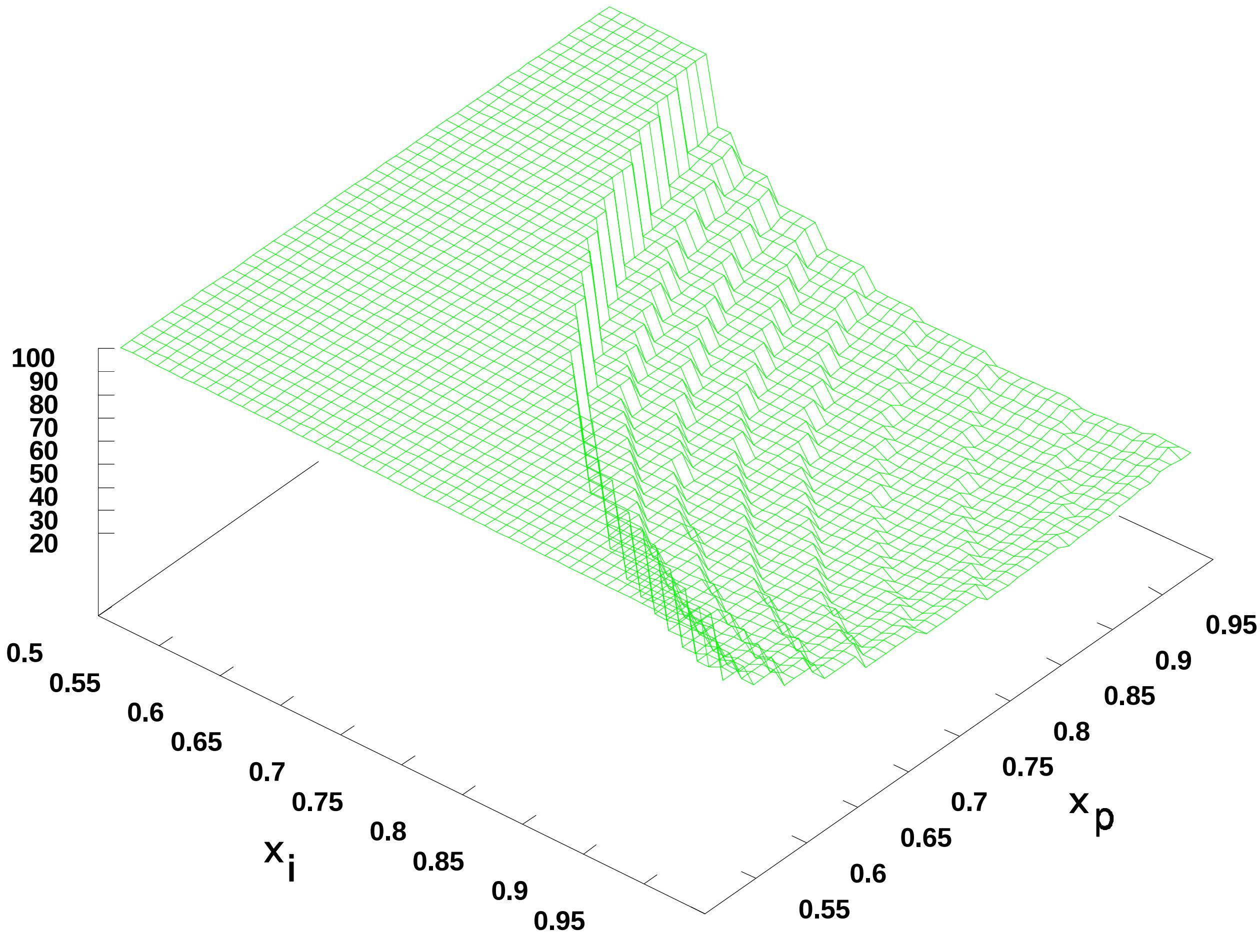

Fig. 13: Objective function for "Red's Nightmare", CASE4, and an overall redundancy of r=10%.

The following table shows the optimized values of x_i, x_p and x_b for the video Red's nightmare and the assumption of an encoding into 100 2kByte packets. Note that this is only an approximation as with changing redundancy r either the packet size or the number of packets must change. Henceforth the packet loss distribution must change (which, however, hasn't been taken into account here). The optimization has been performed by Differential Evolution [9]. Although the local minima being discernible in the objective function surfaces in Figs. 10 -13 don't seem very challenging to an optimization program it has been found that a global optimization procedure is required to escape the traps being set up by the multiple plateaus and tiny local minima. In addition it could be observed from multiple optimization runs that there is obviously not a single global minimum for f but that there are several ones which, however, differ only slightly in their values x_k.

r	5%			10%			15%		
PET-parameters	x_i	x_p	x_b	x_i	x_p	x_b	x_i	x_p	x_b
CASE 1	0.87	0.87	1.	0.81	0.83	0.96	0.77	0.78	0.92
CASE 2	0.86	0.87	1.	0.79	0.86	0.95	0.74	0.80	0.92
CASE 3	0.82	0.9	1.	0.69	0.84	1.	0.67	0.69	1.
CASE 4	0.95			0.91			0.87		

r	20%			25%			30%		
PET-parameters	x_i	x_p	x_b	x_i	x_p	x_b	x_i	x_p	x_b
CASE 1	0.71	0.77	0.88	0.69	0.72	0.85	0.68	0.70	0.81
CASE 2	0.70	0.77	0.89	0.69	0.72	0.85	0.63	0.7	0.83
CASE 3	0.58	0.71	0.96	0.54	0.67	0.94	0.5	0.65	0.91
CASE 4	0.83			0.8			0.77		

Table 2: Optimized Values x_k for different choices of weighting factors using the video Red's Nightmare.

5. Refinements of the Optimization Model

It has already been mentioned that, although the number of error free packets received determines whether all frames of a certain type can be recovered (representing the "all or nothing" strategy), there might still some valuable information in the received packet stream if this number can not be achieved. If we want to include this property into our optimization model the objective function f gets somewhat more intricate. A possible choice for f is

$$f = p_{n,n_i,1} \cdot k_i + \sum_{j_p=1}^{n_p} p_{n,n_p,j_p} \cdot k_{j_p} + \sum_{j_b=1}^{n_b} p_{n,n_b,j_b} \cdot k_{j_b} \qquad (12)$$

with p_{n,n_k,j_k} denoting the probability that j_k frames of type k are lost, provided that n packets were used to encode th GOP and n_k packets are needed to recover all frames of type k. In the following we will have a closer look at the probability p_{n,n_k,j_k}:

Let c_k denote the number of packets which make up a complete frame of type k and let m_k frames of type k be within a GOP. Assume further that l_k packets will be destroyed. Then the scenario being sketched in Fig. 14 represents the situation which is relevant for the computation of p_{n,n_k,j_k}.

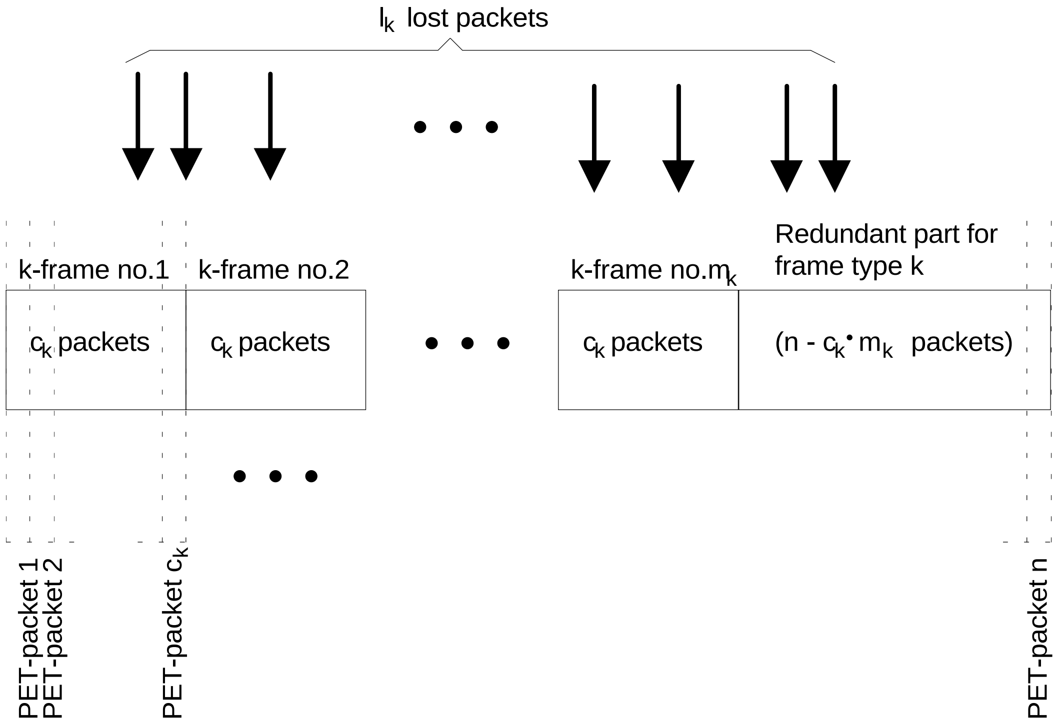

Fig. 14 Packet loss scenario in scheme 2 for a frame of type k.

We compute now the probability that j_k frames of type k are lost. In order to do that it is easier to look at the probability of a frame being unaffected by the l_k destructions. The number of possibilities to distribute l_k "destruction arrows" among n packets is clearly

$$\binom{n+l_k-1}{l_k}.$$

The number of possibilities to distribute l_k destruction arrows among $(n-c_k)$ packets by, for example, leaving out the first c_k packets is

$$\binom{(n-c_k)+l_k-1}{l_k}.$$

Hence the chance of one frame being unaffected by the l_k destructions is

$$P(\text{ 1 frame being unaffected}) = \frac{\binom{(n-c_k)+l_k-1}{l_k}}{\binom{n+l_k-1}{l_k}}. \qquad (13)$$

Using the same line of thought

$$P(\text{ v frames being unaffected}) = \frac{\binom{(n - v \cdot c_k) + l_k - 1}{l_k}}{\binom{n + l_k - 1}{l_k}}. \qquad (14)$$

can be shown to hold. Now we can easily conclude that the chance p_{n,n_k,j_k} that j_k frames are affected

is $$p_{n,n_k,j_k} = P(n_k < l_k \le n - c_k(m_k - j_k)) \cdot P((m_k - j_k) \text{ being unaffected}) \qquad (15)$$

where the first term on the right hand side states that the number of packet destructions l_k must exceed n_k (otherwise all frames can be reconstructed) and that l_k must be small enough to allow for $(m_k - j_k)$ frames to remain unaffected. Taking (14) and (15) into consideration, we finally end up with

$$p_{n,n_k,j_k} = P(n_k < l_k \le n - c_k(m_k - j_k)) \cdot \frac{\binom{n - (m_k - j_k) \cdot c_k + l_k - 1}{l_k}}{\binom{n + l_k - 1}{l_k}}.$$

$$= \sum_{l_k = n_k + 1}^{n - c_k \cdot (m_k - j_k)} P_n(l_k) \cdot \frac{\binom{n - (m_k - j_k) \cdot c_k + l_k - 1}{l_k}}{\binom{n + l_k - 1}{l_k}}, \qquad (16)$$

where $P_n(l_k)$ is the probability of loosing exactly l_k packets provided that a batch of n packets has been sent into the network. Note that we have assumed that the l_k destructions are evenly distributed among the n packets, which might not be true in a bursty environment. However, due to the lack of a reasonable way to model bursty errors, (16) appears to be a sensible approximation.

6. Criticism and suggestions for improvement

Although the optimization model outlined so far makes some sense from a mathematical point of view there are several points of criticism worth mentioning:

1) **MPEG-2**: It makes no sense to squeeze graceful degradation into MPEG-1 because graceful degradation will be implemented in MPEG-2 anyway. As MPEG-2 will also be a standard, MPEG-1 will soon be obsolete.

2) **Objective function:** The entire optimization endeavour thrives on the quality of the objective function. The function set up so far, however, is entirely technical and doesn't account for the human visual system in a sufficient way, although it is exactly this system which is supposed to benefit from PET. In addition, it is not clear at all how to choose the weighting factors in (7) or (12).

3) **Optimization accuracy:** Although the human visual system is very sensitive primarily to temporal resolution errors, first tests indicate that small changes in the values of the PET

parameters cannot be perceived. Therefore it is very likely that a global optimization is not necessary and a much coarser optimization suffices.

4) **Redundancy Levels:** The consideration of just the three frame types of MPEG-1 for PET's multi-level redundancy assignment allows for anything but a video quality degradation that can be called graceful. The number of levels most probably has to be increased and has to go beyond the concept of frame types.

5) **Packet loss distribution:** To do the optimization outlined above the packet loss distribution p_{n,n_k} has to be determined for basically each video stream and the current traffic load. This might be infeasible, especially when the PET parameters shall be adapted in real-time in a scenario where the traffic load is subject to sudden changes.

Let's treat these points of criticism one after the other.

1) **MPEG-2**

It's very likely that MPEG-2 will also become a standard but MPEG-1 is already a standard, and the number of applications that make use of it is continuously on the rise. Even if MPEG-2 should make MPEG-1 obsolete at some time in the future, there will be a lot of applications left that won't be changed from MPEG-1 to MPEG-2. The stubborn presence of FORTRAN and MS-DOS serves as a good example to argue that MPEG-1 will also stay around for quite a while.

2) **Objective function**

From the previous chapters it seems evident that there is no way around subjective tests in order to achieve a pertininent redundancy assignment. Even if more information about the mathematical modeling of video quality is available, the design of the objective function will still be difficult. A possible way to escape the contrivance of a mathematical objective function at all is to use the human being directly as an "objective function gauge". An optimization method which allows for this idea is Experimental Differential Evolution (Experimental DE) as being suggested in [10]. The principal idea is illustrated in Fig. 15.

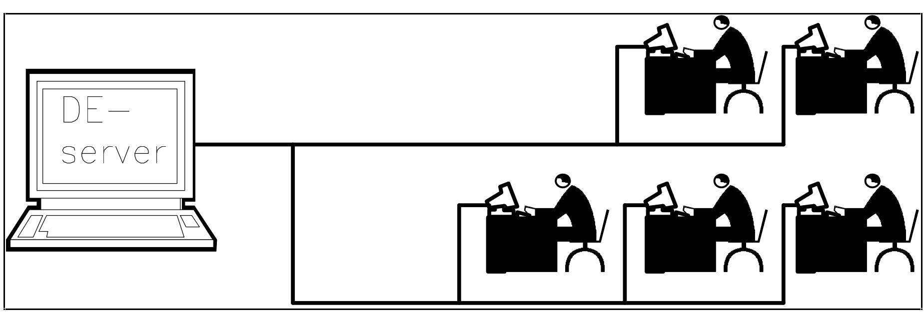

Fig. 15: Scenario for experimental DE to optimize the PET parameters.

A number of people have to watch multiple experiments each of which contains two versions of a predetermined video sequence. The two versions corresond to two different PET parameter settings which have been chosen by a server which works according to the DE strategy. The server tries to present different parameter settings to each person so that it is highly unlikely that two persons watch the exact same version of a video sequence. All the people have to do is to decide which version of the both videos is the better one and tell this decision to the DE-server. The DE-server will choose new parameter settings and present two new video versions to the test persons. While this process continues the parameter settings are optimized by the DE-server based upon the decisions of the test persons. Although this experimental method appears to be outdated in a computerized world it is an effective way to directly involve the human visual system in the optimization process and to circumvent the mathematical modeling of an objective function. DE has been shown to converge smoothly and relatively fast [9]. Therefore it can be assumed that each new experiment yields further results and is not a waste of time due to stagnation behaviour of the optimization algorithm. The optimization can be stopped if no perceivable improvement is visible any more. This could be indicated by a majority decision of the test persons. As DE is a parallel method, the speed of optimization can be increased by employing many persons in parallel and questioning their opinion over a network of computers. This optimization has to be performed for several types of videos and several error scenarios.

3) **Optimization accuracy**

The accuracy problem is solved directly by the experimental DE optimization described above. If most of the test persons agree that no improvement can be perceived any more, the optimization process can be stopped.

4) **Redundancy Levels**

The simplest idea to increase the number of redundancy levels in PET is to make further distinctions between the B-frames. This, however, doesn't have any major impact, especially if there are only two B-frames between P-frames which often is the case [5]. There is probably no way around partial decoding of MPEG-1 and splitting the information in an appropriate way. A suggestion is illustrated in Fig. 16.

The idea behind an enhanced PET scheme as indicated in Fig. 16 is to exploit the knowledge of the human visual system which is already available. Several facts might be important [5]:

1) The human visual system perceives 20 frames per second as smooth motion.

2) Perception is more sensitive to low frequencies which correspond to larger homogeneous areas in a picture.

3) Perception is more sensitive to changes in luminance and along the blue-orange axis than to changes in chrominance and the other colour axes.

4) Human vision emphasizes edge detection.

5) There is visual masking after significant luminance changings, i.e. after a sudden change from a dark scene to a bright scene the first 2 to 3 frames are not perceived.

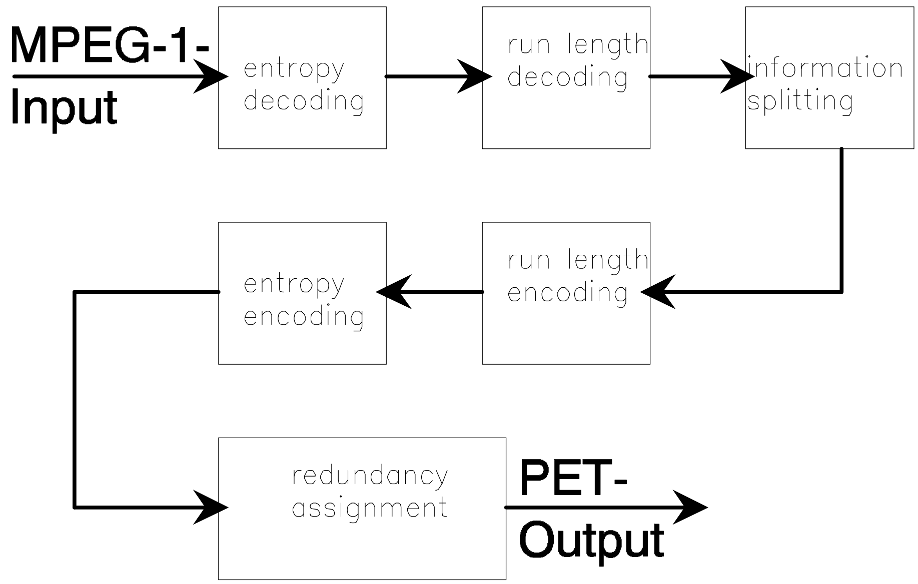

Fig. 16: Future versions of PET might include partial de- and encoding of MPEG-1-streams as well as information splitting.

From the above it might be sensible to split the information in Fig. 16 as suggested in the following (the abbreviations LF and HF stand for low and high frequency respectively):

1) I-frames luminance LF
2) P-frames luminance LF
3) B-frames luminance LF
4) I-frames luminance HF
5) P-frames luminance HF
6) B-frames luminance HF
7) I-frames chrominance
8) P-frames chrominance
9) B-frames chrominance

5) **Packet loss distribution**

Packet networks are highly dynamical systems and permanent changes in packet loss distributions are the rule. Hence it is infeasible to maintain a constant update of the packet loss distribution as measurements to determine the pdf are quite expensive. On the other hand the exact knowledge of the packet loss distribution most probably is neither necessary nor sufficient to provide an adaptive PET scheme that controls its parameter settings according to the network conditions and the video content. If

PET is used in a multicast environment the proper setting of the PET parameters becomes more and more of a compromise anyway as the group members of the multicast scenario might represent strong heterogeneity. An interesting solution could be provided by means of a fuzzy controller as depicted in Fig. 17.

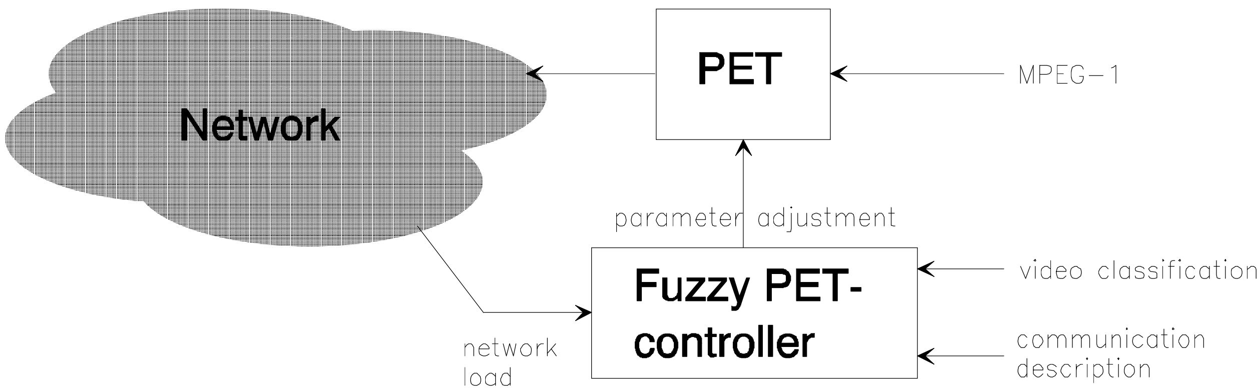

Fig. 17: Adaptive PET via a fuzzy controller.

Fuzzy controllers represent a good controller choice if there is enough knowledge about a system which can't be modeled properly in a mathematical sense. The experimental optimization suggested above will probably yield different optimal parameter settings depending on the video information that is submitted. In a video conference, for example, colour might not be of prime importance whereas in a talk about the life and work of Van Gogh it might well be. Therefore it seems necessary that videos are classified to give the PET system some information on how to set its parameters in an optimal way. Most likely there will be fuzzy rather than crisp classifiers. The same fuzziness could be applicable in the description of the network load and the heterogeneity of the multicast group. The controller could receive information like "an educational video with some entertainment character and a lot of movement" as well as "a moderately loaded network with not too much danger of congestion", and a "roughly homogeneous multicast group". The experience that has to be incorporated into the fuzzy controller could stem, at least partly, from experimental DE optimization.

7. Conclusion

The modeling and optimization of PET redundancy assignment has been investigated. It must be concluded that the purely mathematical approach to solve this problem is questionable to say the least. There are two main reasons for this. The first one is the difficulty to properly map the notion of a good video into a meaningful objective function. The second reason is the necessity to determine the packet loss pdf anew each time there is a significant load change in the network or each time the size and/or number of PET-parameters change. Even if it were possible to use just a few standard pdfs as an approximation the problem of finding the right objective function still remains. As an alternative, an experimental way to solve the optimization problem has been outlined. It has also been suggested to increase the number of redundancy levels and which information content to prioritize in order to possibly enable graceful degradation of MPEG-1 video streams in lossy environments.

8. References

[1] International Computer Science Institute Newsletter, Berkeley, CA, Volume 7, Number 2, Fall 1994.

[2] Chiang, T. and Anastassiou, D., "Hierachical Coding of Digital Television", IEEE Comm. Mag., May 1994, pp. 38 - 45.

[3] Leicher, C., Hierarchical Encoding of MPEG Sequences Using Priority Encoding Transmission, Diplomarbeit am ICSI, 1994.

[4] Wallace, G.K., "The JPEG Still Picture Compression Standard", Comm. ACM, Apr. 1991, Vol. 34, No. 4, pp. 46 - 58.

[5] Rowe, L.A., "Video Compression", Lecture at the University of Berkeley, 1995.

[6] Ismail, M.R., et alii, Modelling Prioritized MPEG Video Using TES and a Frame Spreading Strategy for Transmission in ATM Networks, to be presented at Infocom 95.

[7] LeGall, D., "MPEG: A Video Compression Standard for Multimedia Applications", Comm. ACM, Apr. 1991, Vol. 34, No. 4, pp. 30 - 44.

[8] Albanese, A., Bloemer, J., Edmonds, J. and Luby, M., Priority Encoding Transmission, Technical Report TR-94-039 at ICSI, 1994.

[9] Storn, R. and Price, K., Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012, ICSI.

[10] Storn, R., Optimization of Technical Systems by Differential Evolution, Talk at ICSI, March 13, 1995.