Efficiency of PET and MPEG Encoding for Video Streams: Analytical QoS Evaluations

Bernd E. Wolfinger *

A promising solution in the transmission of video streams via communication networks is to use forward error control in order to mask some of the transmission errors and data losses at the receiving side. The redundancy required, however, to achieve error correction without retransmissions will consume some transmission capacity of a network, therefore possibly enforcing stronger compression of the video stream to be transmitted.

In this paper we introduce analytical models which allow us to determine the expected frame loss probability of MPEG encoded video streams assuming communication via constant bit rate (CBR) virtual circuits with data losses and/or unrecoverable transmission errors. The models can be used to compare the quality-of-service (QoS) as observed on Application Layer for encoding schemes without and with forward error control, possibly making use of different prioritization of transmitted data units (in particular applying PET encoding algorithm as designed at ICSI). The models are applied in various case studies to compare the efficiency of the error control schemes covered.

*) research during sabbatical leave from: Computer Science Department, Telecommunications and Computer Networks Division, Hamburg University, F.R.Germany

CONTENT:

1. Motivation

A well-known principle dating from the still rather early days of data communication is to eliminate redundancy which accidently exists in the user data to be transmitted via a communication network and to replace this original, less useful redundancy by some much more useful redundancy specifically oriented to support forward error control. One could observe this principle e.g. in text transmission, where the considerable redundancy of natural language texts was eliminated by coding algorithms such as Shannon-Fano code and replaced by redundancy being able to recover particularly well from the type of errors to be expected for the underlying transmission channel (e.g. CRC-based checksums [Tan 96]).

This situation in text transmission nowadays seems to be repeated in video communication, where also a lot of redundancy (in space and in time) exists in the video sequences to be transmitted. Video compression algorithms exist (such as MPEG, DVI, ...), which allow one to considerably reduce the redundancy, thereby significantly reducing the throughput required to transmit the compressed new stream. And, as in data communication, researchers and developpers of video communication systems are becoming increasingly aware that video streams after their compression should be transmitted together with some intelligently determined redundancy so that the receiver would be able to correct as many as possible of the transmission errors occured, i.e. to achieve some error control without retransmitting the data (cf. forward error control, FEC). For recent approaches to achieve forward error control in video communication, cf. e.g. [ABE 94], [ILD 94], [ILD 95], [KMS 95], [RiR 95].

A question which arises if one first eliminates redundancy (as inherent, e.g. in a video stream) and thereafter adds redundancy into the stream again, is what kind of efficiency gain would have to be expected as a consequence of the fact that the newly added redundancy is carefully chosen as opposed to the original accidential redundancy. It is typical to approach this question as follows: Suppose compression, add a certain amount of redundancy and then determine the benefit of the additional redundancy. In this paper, however, we want to tackle the question in a somewhat unconventional way, namely: Suppose a certain throughput of a communication network is given, which may be used for transmission of user data as well as for redundant additional data; what is the optimum divisioning of the throughput available for transmitting the user data on the one hand and the additional redundancy to support FEC on the other hand? Note, that compression can be used to reduce the amount of user data to be transmitted; the throughput gained by this compression may then be used to transmit additional redundancy (e.g. for some FEC).

In the following, we are going to compare in detail three classes of coding variants:
- coding *without FEC*,
- coding *with FEC*, but *without prioritizing* the data units transmitted, and, finally,
- coding *with FEC including* a possible *prioritization* of the data units transmitted.

In order to compare between these coding variants (with/ without FEC), we will elaborate on two classes of analytical models (one with FEC and one without), which should allow us to predict the performance and quality-of-service of video encoding algorithms under still sufficiently realistic boundary conditions. As compression algorithm we want to assume the MPEG algorithm [LeG 91], in particular MPEG-1, which in recent times has achieved considerable importance in audio/ video communication. As what concerns FEC we will on the one part consider an FEC algorithm without any prioritization of data units transmitted (could also be seen as one level of priority). On the other part, however, our models reflect the PET coding algorithm [ABE 94], [AlL 96] as developed at International Computer Science Institute (ICSI), which allows one to choose priorities in a very flexible way for the different types of data units transmitted.

To give a very short sketch of MPEG algorithm let us recall that MPEG assumes three types of (video) frames: I-, P- and B-frames. These frames are organized in so-called group of pictures (GOPs). Every GOP starts with an I-frame (intraframe), which is encoded independent of other frames and thus can be displayed independently of the content of other frames which is important to limit effects of error propagation within the total video stream. The other frames in a GOP are P-frames (predictive frames), which are predicted taking into account the content of the last I- or P-frame and thus are dependent on the correctness of those frames. The B-frames (interpolated frames) are bi-directionally interpolated based on the content of the two neighbouring I- resp. P-frames; “neighbouring” here means preceeding or following immediately the group of successive B-frames, where the B-frame to be interpolated is a member of. (The reader is referred to the figures in section 3 for a graphical illustration of typical GOP patterns in MPEG encoded streams and to [LeG 91] for further details on the MPEG standard).

The PET algorithm will be introduced in this paper only to the extend as it is necessary to understand our analytical models as well as the case studies based on those models. The reader interested in details on PET will find such in-depth information e.g. in [ABE 94] and [AlL 96]. The basic approach in PET is to add redundancy r_1, r_2, ..., r_n to the data units d_1, d_2, ..., d_n corresponding to the user data to be transmitted and, then, mapping d_1, d_2, ..., d_n as well as r_1, r_2, ..., r_n onto a (sufficiently large) number of fragments (packets, cells, etc) in a favorable way, e.g. by distributing the complete content of the d_i and the r_j onto the fragments in a uniform manner. This means that, in PET, user data to be transmitted is associated with a sufficient amount of redundancy to let the receiver reconstruct completely the user data as long as having been able to receive a certain percentage (x) of all the data units transmitted without errors/ losses. Evidently, the desire to reduce x implies the need to transmit the data with more redundancy included (i.e. higher overall throughput requirement).

Fig. 1.1 illustrates the situation, which corresponds to the way video communication is assumed to take place throughout this paper and as such is reflected by the models and the case studies. In case that no FEC is applied (variant *(a)*) the digitized video data are compressed according to MPEG and then directly transmitted to the receiver via some a priori established (physical or virtual) circuit; at the receiver, then, decompression takes place, such that the original video stream can be displayed at the receiving site. Of course, transmission errors may have some negative impact on the quality of the stream redisplayed by the receiver. In the case of additional FEC (variant *(b)*), the negative impact of transmission errors is reduced. Here the video data, after its MPEG compression/ encoding, is complemented by redundant information and only thereafter it is transmitted. The receiver evaluates the redundant part of the data received and thus, to a limited extend, can achieve some forward error control. As FEC is executed not by the network but by the distributed application we call the corresponding video-oriented application a “smart application”. In both cases, *(a)* and *(b)*, we assume that the (physical or virtual) circuit used represents a constant bit rate (CBR) connection between sender and receiver, such as it is the case e.g. in communication networks with leased lines or with circuit-switching but also in some of the standardized ATM variants [DeP 96]. Moreover we assume,

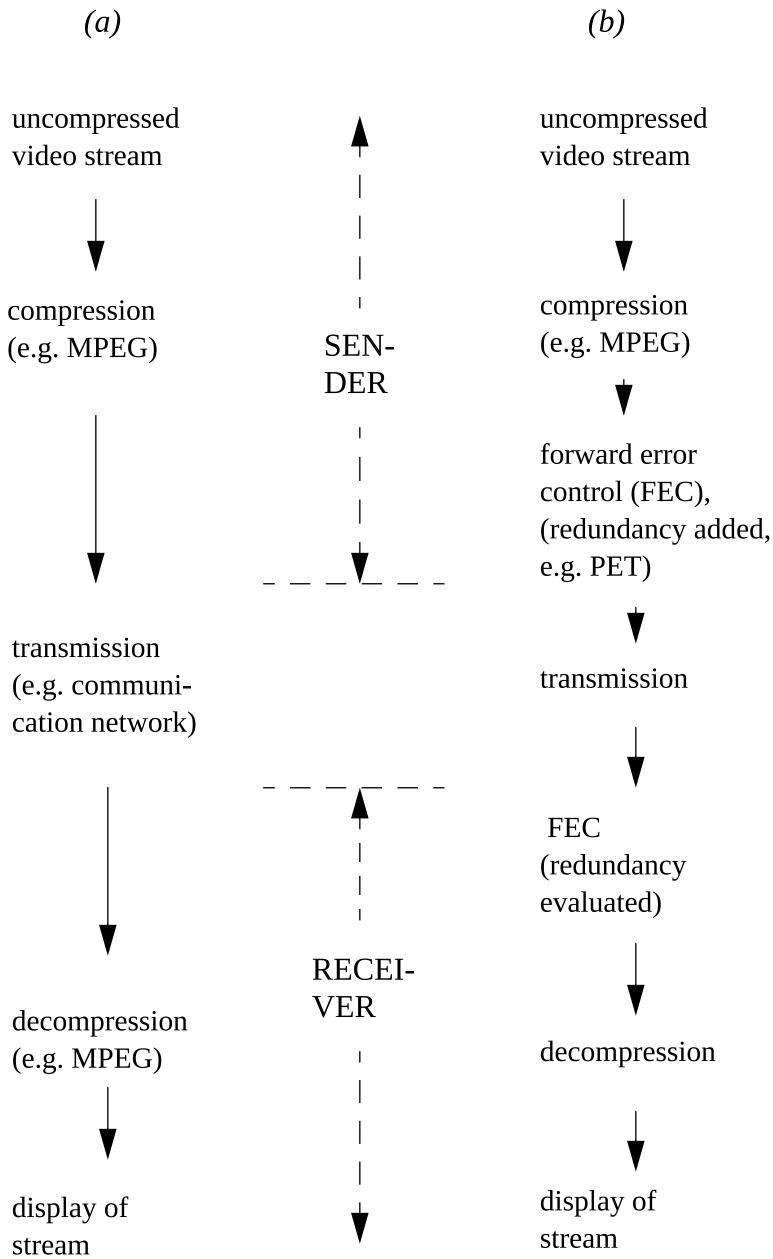

Fig. 1.1: Video communication *without* (a) and *with* (b) forward error control

that the (physical or virtual) circuit used will lose or corrupt data units during their transmission. In case of corrupted data being delivered we suppose that, as a consequence of real-time communication requirements retransmission of the data will not be acceptable. Therefore, we consider data corruption as being equivalent to losses for the receiver as we suppose that on Application Layer no usage will be possible of partially correct data units. So, in the sequel, we only will use the notion of loss, which for us will always include unrecoverable transmission errors and data corruptions. To simplify the formulation in the following, we will also restrict connections as being always virtual circuits (VCs), though by this, we do not want to exclude communication via physically established circuits (temporarily switched or permanent ones), which are still covered by all of our models.

In section 2 we will start with a short summary of the principal assumptions concerning the video communication as viewed throughout this paper. Some justification for the assumptions will be given. We then will introduce our measure of quality (of service) to assess video communication via a network. We conclude section 2 by presenting the resulting model for video communication via unreliable virtual circuits.

Section 3 comprises the analytical model, which allows us to quantitatively evaluate the quality-of-service (QoS) for MPEG encoded video streams via unreliable VCs. A fundamental assumption which allows us a straightforward mathematical/ analytical treatment of the models introduced in section 2 is the mutual independence of losses of data units during their transmission.

Section 4 presents a model similar to the MPEG model which now is oriented towards evaluation of coding variants, which combine MPEG compression with successive encoding to achieve FEC (with or without priorities). The variant with FEC and at the same time allowing prioritization of data units during their transmission reflects the version of PET algorithm as presented in [ABE 94].

In a larger set of case studies (sections 5 - 8) we make intense use of the advantages inherent in our possibility for direct analytical treatment of our models: namely, the excellent usability of analytical models for comprehensive parameter studies. The case studies on the one hand show what kind of investigations are possible in principle with our models. On the other hand, the studies provide also some hints w.r.t. a model based QoS management [DeM 95] to support high quality video communication in real-time. Our goals in the case studies are e.g. investigation of the implications of the amount of available transmission capacity on the quality of coding algorithms (section 5); investigation of the implications of a varying network reliability (section 6); determination of adequate parametrization during data transmission (in particular, search for “optimum” packet size and/ or amount of redundancy to be used during transmission) as a function of throughput available and of the loss rate observed, e.g., by means of measurements (cf. section 7). Some of the case studies assume an underlying packet-switched network, other ones are carried out for a cell-switched network (such as an ATM network).

Section 8 puts together several experimental results by way of example, which among others illustrate the significant performance gains achievable by means of FEC. The section concludes in demonstrating how analyses as those of our case studies might be embedded in a system for model based QoS management.

Section 9 summarizes some of the lessons learned from the case studies carried out up to now and it indicates some of the limitations of the modeling approach as followed by this paper. We sketch some desirable extensions and/ or generalizations for the models used by us. Additional case studies as being planned for the near future are also shortly commented as well as the steps being planned by us in order to obtain a more comprehensive model validation.

2. A Model for Video Communication via Unreliable Networks

In this section we are going to motivate and present the model chosen by us to reflect both cases of encoding, i.e. *without* and *with* forward error control. In addition, we will summarize the main model assumptions and give some justification for them.

Our goal in modeling has been to find adequate models which are relatively simple and still analytically tractable, in order to be used for the purpose of QoS management. Simulation models, although having the potential to be more realistic, according to our opinion could hardly be embedded into a system which has to provide QoS management decisions on a fine grain time scale (e.g. in the order of *msec*). This means that also the evaluation of our models should be as straightforward as possible.

The more restrictive model assumptions typical for analytical models, as opposed to simulative ones, should however be carefully justified and be sufficiently realistic in order to make the resulting models applicable at least for some classes of existing network and load situations (cf. strong impact of load models on validity of network models as emphasized, e.g., in [WoK 90]).

The basic motivation for the models has been to allow us the study of the following question of interest:

"Assuming CBR virtual circuits/ physical channels with transmission errors, how can the constant transmission capacity be optimally used, e.g. in the case of video encoding :

- *without* FEC ---> advantage: less compression is necessary to transmit the stream
- *with* FEC ---> advantage: some losses of the network may be corrected by a "smart application" at the receiving endsystem (also of interest: what is the potential gain of using priorities in FEC over not prioritizing transmitted data units) ?"

In order to get some quantitative assessment of coding variants we have to accept some quantifiable *measure of quality* (i.e. some "quality of service indicator"). In our studies we are going to base our QoS evaluations on the well-known measure:
ε_f = probability of losing any frame in the total video stream to be displayed (including *direct* and *indirect* losses, where direct losses are losses in the network and indirect losses are losses in redisplaying the video stream at the receiving side, where the frame F to be displayed has not been directly lost but a frame on which F depends).

The following *basic assumptions* are underlying our models:

(a) *Losses* of data:

- losses of data for a given VC are allowed to occur in the network (e.g. in wireless links or in packet / cell switches) or in the endsystems (e.g. implied by smoothing problems in a sending endsystem);
- cell/ packet losses, fragment losses for short, in the network or in an endsystem are assumed to be mutually independent (for the subset of data transferred via one specific VC);

- partial loss within a frame (e.g. lost cell) makes the complete frame worthless for the receiver.

(b) *QoS-measure*:

We assume that ε_f is an adequate QoS indicator (i.e. the optimization criterion we therefore will apply).

(c) Dependency of *frame lengths* on GOP pattern:

Finally, we suppose that only a minor dependency of average frame lengths exists for I-, P-, B-frames on the GOP pattern applied for stream encoding; so our test w.r.t. throughput required to transmit a video stream encoded according to a specific GOP pattern will be based only on mean lengths as measured for I-, P-, B-frames, such as in [WoB 95]. (This assumption is only relevant for the case studies in sections 5-8; the models themselves could make direct use of frame length measurements for a given stream and given encoding pattern and we could thus eliminate this assumption, if considered to be too unrealistic for some types of video streams).

The assumptions as stated above are, as is typical for model assumptions, only partially fulfilled in existing communication networks. However, we think that at least in some networks they are still a sufficiently realistic approximation of reality. To substantiate this, let us shortly comment on the validity of our assumptions looking at typical networks and on typical load situations in video communication.

The independence assumption concerning fragment losses is a rather approximate one in a large number of networks. However, one should keep in mind that we need the independence assumption only for losses of fragments transported via one VC and not for all the data units transported by the communication network or switched by one switching node. This makes our independence assumption much more acceptable than e.g. some independence assumptions of other authors supposing independent losses for bit errors etc. Moreover, a distribution of data losses in a network will strongly vary over time and thus typically could not be determined, even not by measurements during network operation. For a method to determine by measurements the cell loss probability (not the distribution!) in ATM networks, cf. [ZhF 96]. For measurements to determine error characteristics for wireless LAN environments, cf. [EcS 96]. Last not least, the losses we assume to happen primarily in the network (within a VC), could evidently be generalized to also include losses in the endsystems, if sender and receiver are not able to determine where losses take place and if losses in the endsystems still can be seen as being sufficiently independent on each other.

The assumption that partial loss of frames will always imply loss of the complete frame will be valid for endsystems which, because of high expenditure (e.g. in computation) of reconstructing partially lost frames, are unable to make use of partially correct frames. For endsystems which are able to correct partially corrupted frames (e.g. on “macro block level” [LeG 91]) our models would only represent an approximation; simulation models would allow a corresponding refinement.

The QoS measure ε_f is not really able to completely reflect QoS quality as observed by an end-user; nevertheless ε_f reflects at least “clusters” of frame losses in the way that a lost cluster of n frames is weighted correspondingly (i.e. with factor n) by ε_f. The metrics which would reflect video quality as observed by human endusers have to take into account human perception and thus are strongly dependent on the content of the video stream transmitted and therefore in conflict to our modeling approach (those more elaborated QoS measures could be usable, to some extend, as part of a simulation model).

The assumption that lengths of I-, P- and B-frames do not vary too much is valid at least for video streams with low intensity of motion in the stream. Load measurements for such video streams show [Bai 96] that the type specific length variation for different types of frames stay well between 10-20% deviation of the average frame lengths (at least during time intervals of a few seconds as they are typical for the QoS management we want to support by our models).

To conclude this section let us summarize our model assumed for video communication, before attacking the problem of analytical model evaluation in sections 3 and 4. Our models take into account the following proceeding to realize a video communication:

- 1. Uncompressed video stream (V_u) is produced at sending endsystem
- 2. Video stream V_u gets MPEG encoded (let V_M denote the result)
- 3. V_M is processed further by the sender ---> Variants:
  3.a) Variant I (*without* FEC): V_M as such is forwarded to fragmentation process
  3.b) Variant II (*with* FEC): redundancy is added to V_M (*with* or *without* priorities) using PET ---> resulting stream be V_P ; V_P is forwarded to fragmentation process
- 4. Fragmentation of V_M resp. V_P (into cells or packets)
- 5. Transmission of fragments using CBR-VC (fragments lost according to loss probability assumed)
- 6. Reassembly of frames at receiving endsystem
- 7. Decoding of video stream (possibly after FEC with variant II)
- 8. Display of video stream (frame F is missing, if F itself has been (partially) distroyed during transmission, called *“direct loss”* in the sequel, or at least one frame has been distroyed on which F depends, called *“indirect loss”*).

The models will now be evaluated under the model assumptions stated above, in the way that section 3 covers MPEG encoding only (i.e. no FEC) and section 4 covers MPEG encoding combined with FEC (both with and without priorities for transmitted data units).

3. Analytical QoS Evaluation for MPEG Encoded Video Streams

In order to derive the mathematical results required to determine ε_f, i.e. our QoS measure of interest, let us introduce the following notations, abbreviations, etc:

MPEG:

- GOP: group of pictures (for MPEG)
- (N,M): MPEG parameters to characterize a GOP pattern
  N: distance between neighbouring I-frames
  M: distance between I-frame and next I-/P-frame
- n_i , n_P , n_B: number of I-/P-/B-frames in a single GOP; evidently, $n_i = 1$
- |GOP| : total number of frames in a single GOP (i.e. $|GOP| = n_i + n_P + n_B = N$)

PET:

- r: percentage of redundancy spent for PET encoding (in [%])
- x_i , x_P , x_B: fraction of cells/ packets required to reconstruct all I-/P-/B-frames resp.
  ---> cf. PET parameters as introduced, e.g. in [Sto 95]; in particular: $x_i \le x_P \le x_B$;
  note: $x_i = x_P = x_B$, if no priorities for different types of frames are used in forward error control

Virtual Circuit (VC):

- d: (constant) data rate transmitted by VC, (in [kbit/s])
- ε_c: cell/ packet loss probability for VC
- l_c: constant cell size resp. maximum packet size, (in [Byte])
- h_c: size of cell resp. packet header, constant (in [Byte])
- l_0: average frame size acceptable taking into account d and ν (cf. below), (in [Byte])

Video Stream:

- ν: video display frequency, (in [Hz]); let T denote: $T=1/\nu$
- $l_i(N,M)$, $l_P(N,M)$, $l_B(N,M)$: mean length of I-/P-/B-frames for a given video stream after MPEG encoding (using N,M as MPEG parameters), (in [Byte])
- c_i , c_P , c_B: mean number of cells/ packets required to transmit I-/P-/B-frames via VC for a given video stream and a given GOP pattern (including PET specific redundancy if FEC is applied)

Losses:

- ε_i , ε_P , ε_B : loss probability for an I-/P-/B-frame of average size (for a given stream); refers to direct losses only, i.e. to losses in network resp. during frame transmission
- n_ε : mean number of losses to be expected in a single GOP (for a given GOP pattern) as a result of direct and indirect frame losses.

We are now able to present the

Basic algorithm *for calculation of* ε_f *(Variant:* ***MPEG only****, i.e.* ***no FEC****)*

- **STEP1:** Determine the set M_{NP} of possible GOP patterns, which can be applied for the *fixed* video stream taken into account data rate (d) of VC; assumption: $|GOP| \leq 10$ (because of the delay resulting of large GOP sizes, making real-time communication impossible).
- **STEP2:** For every coding pattern $M_c \in M_{NP}$ repeat:
  STEP2a: Determine ε_i , ε_P , ε_B
  STEP2b: Determine n_ε
  STEP2c: Determine $\varepsilon_f = \varepsilon_f(M_c)$.
- **STEP3:** Determine optimum GOP pattern M_c*, i.e. calculate $\varepsilon_f^* = \min \{\varepsilon_f(M_c) \mid M_c \in M_{NP}\}$
  ---> M_c* corresponds to GOP pattern associated to ε_f^* .

This algorithm allows us to determine the QoS measure ε_f, namely the mean frame loss probability (direct and indirect frame losses taken into account) for total video stream under the assumptions stated in sections 1 and 2.

In the following let us now refine the three principle steps of our basic algorithm.

- **cf. STEP1:** *Determination of possible GOP patterns*

To find out whether a given GOP pattern can be used with the throughput available (d) we have to test whether the corresponding MPEG encoding would lead to a throughput requirement still less or equal to d. This test is done by the following algorithm (“Throughput Test”) which assumes that variations in the frame lengths during the stream can be eliminated still within the same GOP with a sufficiently high probability just by means of smoothing at the sender.

- **cf. STEP2:** *Calculation of overall frame loss probability*

The frame loss probability is calculated (in STEP2c) based on the fragment loss probabilities given and their impact on the probability for direct losses of I-, P- and B-frames (in STEP2a).

STEP2a: Independence assumption of cell/ packet losses implies:

$$\varepsilon_i = 1 - \left(1 - \varepsilon_c\right)^{c_i}$$

as the loss probability of average size I-frames. (Corresponding results for P- and B-frames; only difference: P- and B-frames may not exist in GOP, then, of course, $\varepsilon_P = 0$ or $\varepsilon_B = 0$).

STEP2b:

1. GOP structure (example):

By way of example let us first shortly illustrate the implications of direct losses of frames on a given GOP (here GOP_x).

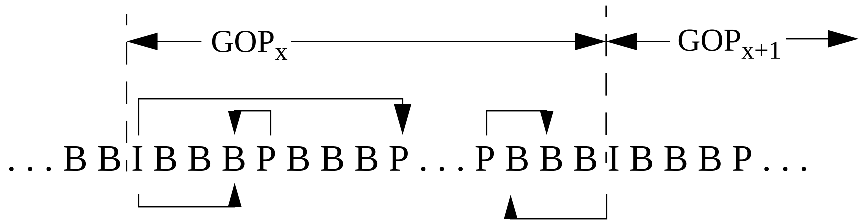

Fig. 3.1: Implications of I- and P-frame losses

<u>Throughput Test</u>

$c_i = \lceil l_i / (l_c - h_c) \rceil$; $c_P = \lceil l_P / (l_c - h_c) \rceil$; $c_B = \lceil l_B / (l_c - h_c) \rceil$;

/* $\lceil x \rceil$: "ceiling function" */

i = 0;

l_0 = (d*1000/8)/v; /* determine average frame size acceptable */

for j = 1, 2, ... , m_g *do* pattern_found[j] = false;

/* m_g = maximum size acceptable for GOP, m_g = 10 */

p = 0; /* loop for varying the number of P-frames; p = number of P-frames assumed in GOP during this iteration */

NEW_P_SET: M = 1;

/* loop for successively increasing the number of B-frames */

NEW_B_SET: N = M*(p+1);

/* calculate MPEG parameter N; number of B-frames just modified */

b = N-p-1; /* calculate new number of B-frames */

if N > m_g *then goto* PATTERN_FOUND_FOR_THIS_P;

if cell-switching *then*

THROUGHPUT_REQ = (c_i + p*c_P+ b*c_B)*l_c

else /* packet-switching */

THROUGHPUT_REQ = (l_i + p*l_P+ b*l_B) + (c_i + p*c_P+ b*c_B)*h_c ;

/* THROUGHPUT_REQ: data to be transmitted in N*T */

if THROUGHPUT_REQ > N * l_0 *then goto* CONT_B_LOOP;

/* here the left-hand-side of the comparison test corresponds to the required and the right-hand-side to the available throughput during time-interval N*T (i.e. period corresponding to the duration of a GOP) */

pattern_found[p] = true; /* pattern found for this number of B-frames */

MPEG_PATTERN[i] = (N,M); /* save MPEG parameters of this GOP pattern */

i = i+1;

CONT_B_LOOP: M = M+1; *if* N < m_g *then goto* NEW_B_SET;

PATTERN_FOUND_FOR_THIS_P: *if* NOT pattern_found[p] *then goto* EXIT;

p= p+1; *if* p < (m_g-1) *then goto* NEW_P_SET;

EXIT: *end*;

2. We now want to summarize what kind of losses would “hurt” GOP_x . In order to make sure that we count every loss of a frame exactly once, it is convenient to assume the following “Loss Hierarchy”:

(a) Losing the I-frame of GOP_x ,

(b) Losing any P-frame of GOP_x ,

(c) Losing any B-frame of GOP_x ,

(d) Losing the I-frame of GOP_{x+1} .

This loss hierarchy is to be interpreted as follows: A frame loss F_l is always considered as being a consequence of that loss which occurred highest in the above hierarchy, in the case that loss F_l would be implied at the same time by two or more other losses.

We are now prepared to discuss the implications of frame losses according to the loss hierarchy which we want to respect:

(a) <u>I-frame of GOP_x lost</u>

Consequence of direct loss of I-Frame of GOP_x :

N frames lost (directly or indirectly), i.e. the expected impact of (a) on GOP_x implies:

$$n_{\varepsilon, I1} = \varepsilon_i \times N$$

where $n_{\varepsilon,I1}$: expected number of lost frames in a single GOP being consequence of having lost the I-frame of this GOP.

(b) <u>P-frame of GOP_x lost</u>

Consequence of direct loss of P-frame (P*) of GOP_x :

- this P-frame is lost,
- the group of consecutive B-frames (*B-Group* for short) left of P* is lost,
- the B-Groups and the P-frames to the right of P* are lost.

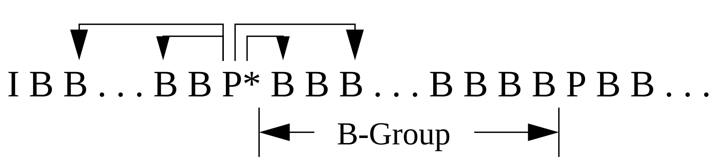

Fig. 3.2: Implications of P-frame losses

note:

(1) all these losses would already have been occurred if the I-frame of GOP_x would have been lost;

(2) in addition, we have to make sure that we count also those indirect B-frame losses only once, where, e.g., at least one P-frame left of a lost B-Group B_G* and the P-frame immediately right of B_G* would have been (directly) lost.

Convention:

Let us count indirect losses of a B-Group B_G* (resulting from P-frame losses) such that the P-frame most to the left of the GOP is considered to be responsible for the indirect loss of B_G* (if more than one P-frame had been lost at the same time and thus had implied the indirect loss of B_G*).

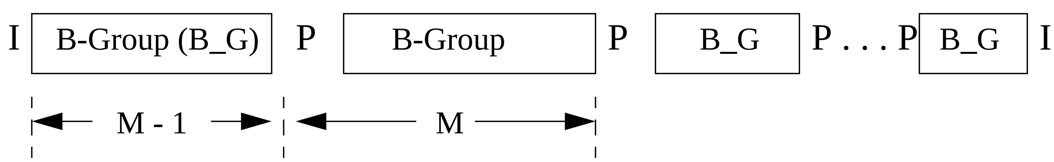

Fig. 3.3: Implications of P-frame losses onto different B-Groups within a GOP

Effect of losing first P-frame of GOP_x:

- *left-side* impact: M - 1 losses with probability

$$\varepsilon_P \times \left(1 - \varepsilon_i\right)$$

- *right-side* impact: $M*n_P$ losses with probability

$$\varepsilon_P \times \left(1 - \varepsilon_i\right)$$

Effect of losing j-th P-frame (j>1) of GOP_x:

- *left-side* impact: M - 1 losses with probability

$$\varepsilon_P \times \left(1 - \varepsilon_i\right) \times \left(1 - \varepsilon_P\right)^{j-1}$$

- *right-side* impact: $M*(n_P-j+1)$ losses with probability

$$\varepsilon_P \times \left(1 - \varepsilon_i\right) \times \left(1 - \varepsilon_P\right)^{j-1}$$

Overall result:

$$n_{\varepsilon,P} = \sum_{j=1}^{n_P} \left((M-1) \times \varepsilon_P \times \left(1-\varepsilon_i\right) \times \left(1-\varepsilon_P\right)^{j-1} + M \times \left(n_P + 1 - j\right) \times \varepsilon_P \times \left(1-\varepsilon_i\right) \times \left(1-\varepsilon_P\right)^{j-1} \right)$$

$$= \varepsilon_P \times \left(1-\varepsilon_i\right) \times \sum_{k=0}^{n_P-1} \left(\left(M - 1 + M \times \left(n_P - k\right)\right) \times \left(1-\varepsilon_P\right)^{k} \right)$$

where $n_{\varepsilon,P}$: expected number of lost frames in a single GOP being consequence of having lost any P-frame of this GOP (but not yet the I-frame);

note also that $n_P = N/M-1$.

(c) <u>B-frame of GOP_x lost</u>

A direct loss of a B-frame will contribute to a new loss (not yet counted as a consequence of a possibly lost I- or P-frame):

- for B-frames of the first B-Group with probability:

$$\left(1-\varepsilon_i\right) \times \left(1-\varepsilon_P\right)$$

---> I-frame *and* first P-frame not lost;

- for B-frames of the B-Group B_G_j not being first or last with probability:

$$\left(1-\varepsilon_i\right) \times \left(1-\varepsilon_P\right)^{j-1} \times \left(1-\varepsilon_P\right)^{1}$$

---> I-frame *and* all P-frames left of B_G_j *and* P-frame directly right of B_G_j not lost;

- for B-frames of the last B-Group B_G_{np+1} with probability:

$$\left(1-\varepsilon_i\right) \times \left(1-\varepsilon_P\right)^{n_P}$$

---> I-frame *and* all P-frames left of B_G_{np+1} not lost.

If we take into account that the size of each B-Group is:

M - 1,

we can conclude that

$$n_{\varepsilon,B} = (M-1)\sum_{j=1}^{n_P}\left(\varepsilon_B \times (1-\varepsilon_i) \times (1-\varepsilon_P)^j\right) + (M-1) \times \varepsilon_B \times (1-\varepsilon_i) \times (1-\varepsilon_P)^{n_P}$$

where $n_{\varepsilon,B}$: expected number of lost frames in a single GOP being consequence of having lost any B-frame of this GOP (but not yet the I-frame or any P-frame).

(d) <u>I-frame of GOP_{x+1} lost</u>

Loss of I-frame of GOP_{x+1} affects just last B-Group of GOP_x.

Those B-frames have already been considered to be lost, if:

- preceeding I-frame (of GOP_x),
- at least one P-frame (of GOP_x), or
- the B-frame itself

had been lost.

Thus,

$$n_{\varepsilon,I2} = \varepsilon_i \times (1-\varepsilon_i) \times (1-\varepsilon_P)^{n_P} \times (1-\varepsilon_B) \times (M-1)$$

where $n_{\varepsilon,I2}$: expected number of lost frames in a single GOP_x being consequence of having lost the I-frame of the following GOP_{x+1} (but not yet having been lost already as a consequence of any direct frame losses within GOP_x).

To conclude STEP2b we have to calculate

$$n_\varepsilon = n_{\varepsilon,I1} + n_{\varepsilon,P} \times \delta_P + n_{\varepsilon,B} \times \delta_B + n_{\varepsilon,I2} \times \delta_B$$

where $\delta_P = 1$ if $n_P > 0$ and $\delta_P = 0$ if $n_P = 0$ (δ_B defined correspondingly).

cf. STEP2c: $\varepsilon_f = n_\varepsilon / N$.

- **cf. STEP3:** *Determination of optimum GOP pattern*

This step is straightforward, thus the analytical QoS evaluation for the case without FEC is concluded.

4. Analytical QoS Evaluation for MPEG/PET Encoded Video Streams

Let the notation as introduced in section 3. still hold.

Again, we start our discussion to derive the results required to achieve analytical evaluation of our models, covering now FEC, with the

***Basic algorithm** for calculation of ε_f (Variant: **MPEG/PET**, i.e. **FEC applied**)*

- **STEP1:** Determination of parameters for the coding algorithm applied and of possible GOP patterns
  STEP1a: Choose percentage r of redundancy used for FEC and choose priorities of FEC for different frame types (I, P, B), i.e. decision of values for x_i , x_P , x_B (cf. PET algorithm)
  STEP1b: Determine set M_P of possible GOP patterns, which can be applied for the *fixed* video stream taken into account data rate (d) of VC and redundancy used for FEC; again assumption: $|GOP| \leq 10$.
- **STEP2:** For every coding pattern $M_c \in M_P$ repeat:
  STEP2a: Determine probability of losing $\geq$ k fragments within a GOP
  STEP2b: Determine n_ε
  STEP2c: Determine $\varepsilon_f = \varepsilon_f(M_c)$.
- **STEP3:** Determine optimum GOP pattern M_c*, i.e. calculate $\varepsilon_f^* = \min \{\varepsilon_f(M_c) \mid M_c \in M_P\}$
  ---> M_c* corresponds to GOP pattern associated to ε_f^* .

Again, as in section 3, we want to refine the principle steps of our basic algorithm.

- **cf. STEP1:** *Determination of parameters for the coding algorithm applied and of possible GOP patterns*

cf. STEP1a:

This step is new as compared to the MPEG-oriented algorithm in section 3. The percentage r of redundancy used and the prioritization of frame types has to be chosen by the experimenter (thus it represents input data for model evaluation).

cf. STEP1b:

This step corresponds to STEP1 of the MPEG-oriented algorithm in section 3 with just one (minor) modification:

Calculation of THROUGHPUT_REQ has to be replaced by:

if cell-switching *then*
THROUGHPUT_REQ = $\lceil ((l_i + p{*}l_P + b{*}l_B){*}(1+r)) / (l_c - h_c) \rceil {*} l_c$
else /* packet-switching */
THROUGHPUT_REQ = $(l_i + p{*}l_P + b{*}l_B){*}(1+r) +$
$\lceil ((l_i + p{*}l_P + b{*}l_B){*}(1+r)) / (p_l - h_c) \rceil {*} h_c$;

note:

(a) Here p_l stands for the (constant) size of packets used for transmission (with $p_l \leq l_c$); in PET encoding it will be desirable in some situations to use packets with smaller than the maximum size possible (cf. experimental results in several of the case studies).

(b) The set of acceptable GOP patterns is now denoted by M_p (as opposed to M_{NP} as in section 3.)

- **cf. STEP2:** *Calculation of overall frame loss probability*

Steps 2a and 2b become significantly different from MPEG-oriented algorithm due to characteristics of FEC in PET algorithm.

cf. STEP2a:

The mean number of fragments n_c to transport all data (including redundancy) corresponding to a GOP is

$$n_c = \left\lceil \left(\left(l_i + p \times l_P + b \times l_B \right) \times (1 + r) \right) / l_c \right\rceil$$

(cf. STEP1b.)

Due to the independence assumption of fragment (e.g. cell) losses (occurring with probability ε_c) the probability $p_{\geq x}$ of losing at least x fragments in n_c successively transmitted fragments is

$$p_{\geq x} = \sum_{j = x}^{n_c} \left(\binom{n_c}{j} \times \varepsilon_c^j \times \left(1 - \varepsilon_c \right)^{n_c - j} \right)$$

cf. STEP2b:

The minimum number of fragments missing which eliminate an I-frame is

$$z_i = \left\lfloor \left(1 - x_i \right) \times n_c \right\rfloor + 1$$

where $\lfloor x \rfloor$ denotes the “floor function”.

Correspondingly, if

$$z_P = \left\lfloor \left(1 - x_P \right) \times n_c \right\rfloor + 1$$

or more fragments are missing, all P-frames are lost (analogous result for z_B).

Thus, chosing the same hierarchy of loss implications as in section 3, i.e.

- direct loss of I-frame of GOP_x ,
- direct loss of P-frame (here all P-frames) of GOP_x ,
- direct loss of B-frame (here all B-frames) of GOP_x ,
- direct loss of I-frame of GOP_{x+1} ,

now yields

$$\begin{aligned} n_\varepsilon = {} & N \times f(\varepsilon_c, z_i, n_c) + \\ & (N-1) \times f(\varepsilon_c, z_P, z_i - 1) \times \delta_P + \\ & (N - N/M) \times f(\varepsilon_c, z_B, z_{P/i} - 1) \times \delta_B + \\ & (M-1) \times f(\varepsilon_c, z_i, n_c) \times \left(1 - f(\varepsilon_c, z_B, n_c)\right) \times \delta_B \end{aligned} \qquad (eq.*),$$

where $z_{P/i} = z_P$ (if $n_P > 0$) and
$z_{P/i} = z_i$ (if $n_P = 0$),

$$f(p, m, n) = 0, \; if \; (p = 0) \; OR \; (m > n) \; OR \; ((p = 1) \; AND \; (n \neq n_c)) \quad \text{and}$$

$$f(p, m, n) = 1, \; if \; (p = 1) \; AND \; (n = n_c)$$

$$f(p, m, n) = \sum_{j=m}^{n} \binom{n_c}{j} \times p^j \times (1-p)^{n_c - j} \quad \text{, otherwise}$$

Note that every line of the eq. *(eq.*)* corresponds to the implication of the corresponding level of the loss hierarchy. Again, as in section 3 we have to make sure that we count losses only once (conditional probabilities).

STEP2c and 3 remain unchanged as compared to the MPEG oriented algorithm, in particular:

$\varepsilon_f = n_\varepsilon / N$

still holds for the case of MPEG/PET encoding.

It should be evident that the algorithm given also covers the case of **FEC without priorities**. In this case, *(eq.*)* simplifies to

$$\begin{aligned} n_\varepsilon = {} & N \times f(\varepsilon_c, z, n_c) + \\ & (M-1) \times f(\varepsilon_c, z, n_c) \times \left(1 - f(\varepsilon_c, z, n_c)\right) \times \delta_B \end{aligned}$$

as now $z_i = z_P = z_B = const = z$.

Thus, we have concluded our analytical QoS evaluations for MPEG based coding algorithms of video streams without and with forward error control (possibly applying priorities for transmission of video frames).

In the following we want to look at various case studies in order to quantitatively compare the quality-of-service achieved by each coding variant and thus investigating the potential of “smart (distributed) applications”.

5. Case Study I: Implications of Available Transmission Capacity

The purpose of the case studies in the subsequent sections is twofold: first of all, they have been conceived to quantitatively evaluate and to compare the QoS which is achievable by different variants of video encoding; secondly, the studies are meant to illustrate the areas of applicability (e.g. in the context of a model-based QoS management) of the analytical models presented in the first part of this paper.

Our goal for *case study I* has been, to investigate how the available transmission capacity of the VC used for the video communication influences the video quality (for the different encoding schemes considered). Evidently, reduced transmission capacity will make usage of some MPEG coding patterns (GOP patterns) impossible, which could have a negative impact on the video quality, because the patterns lost by capacity reductions could have been the optimum ones. In addition, it is interesting to see at which transmission capacity no GOP pattern at all would be left for video encoding, respecting the throughput limitation imposed on the VC.

The boundary conditions chosen for case study I have been as follows:

- Packet-oriented communication using a VC transmitting data at a rate $d \leq 128$ kb/s (constant bit rate available was assumed in each experiment), note: 128 kb/s correspond e.g. to 2 B-channels in ISDN (narrow-band) [Tan 96]; 10 Byte control information per packet; maximum packet length: 1024 Byte.
- A video stream corresponding to the video Red's Nightmare as also considered by [Sto 95] and leading to values $l_i = 1367$ Byte, $l_P = 900$ Byte, $l_B = 250$ Byte for mean frame sizes (values correspond to 1/9 of values/ measurement results given in [Sto 95]; generation and transmission of smaller frames required as a consequence of the rather low data rate); video display frequency $\nu = 30$ Hz and $|GOP| \leq 10$.

We varied throughout case study I:

- the packet loss probability ε_c (assumed to be independent of the packet length!),
- the data rate d (varied in steps of 5 kb/s),
- the amount of redundancy spent for forward error control,
- the prioritization of I-, P-, B-frames in case of FEC with priorities.

The encoding variants as observed in case study I were:

- *MPEG*: encoding according to MPEG-1 without any FEC;
- *FEC(r, p_l)*: forward error control without priorities, using r % of redundancy and a packet-length of p_l (values used: $r \in \{0.05, 0.1, 0.2, 0.3\}$ and $p_l \in \{128, 512\}$); note: we write *FEC(r, *)* if value for p_l is unspecified;

- *pFEC(r, p_l)*: forward error control with priorities, semantic of r and p_l as in FEC(r, p_l), use of redundancy such that for
  r = 0.05: $x_i = x_P = 0.87$, $x_B = 1.0$,
  r = 0.1: $x_i = 0.79$, $x_P = 0.86$, $x_B = 0.95$,
  r = 0.2: $x_i = 0.71$, $x_P = 0.77$, $x_B = 0.88$,
  r = 0.3: $x_i = 0.68$, $x_P = 0.7$, $x_B = 0.81$,
  (cf. values for use of r as suggested by [Sto 95]).

Fig. 5.1 shows, for a very high packet loss probability of $\varepsilon_c = 0.01$, what quality-of-service the encoding variants observed are able to achieve (ε_f * denotes the optimum value of ε_f , optimized over all GOP patterns available for the encoding variant under the given throughput limitation). The results show that, as is to be expected, for encoding without FEC or with small r, no acceptable quality can be achieved as a consequence of the high packet loss probability supposed. The results also demonstrate that with sufficiently large r a still good overall video quality could be expected, if packet losses are mutually independent as assumed in the models. In Fig. 5.1 it is also illustrated that no usable GOP pattern will be left in cases where too much redundancy for FEC is used (as compared to the available data rate d) or the data rate of the VC is too strongly reduced (given the fixed amount of redundancy).

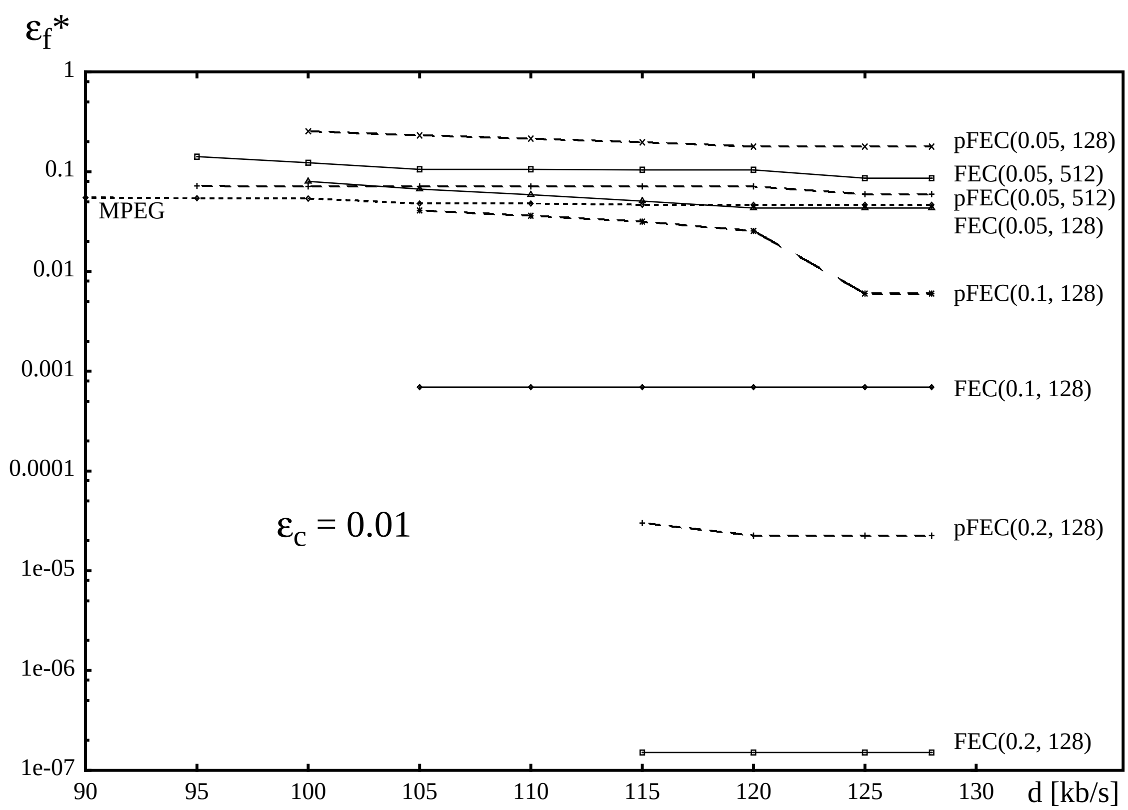

Fig. 5.1: Frame loss probability for and principle usability of different encoding variants in dependence of the available transmission capacity (coarse-grain view) (in case of r = 0.3: ε_f* << 10^{-8} for all d, thus, those results are not covered by Fig. 5.1)

Fig. 5.2, for $\varepsilon_c = 0.001$, is a refined illustration of the fact that encoding variants will lose their optimum GOP patterns (leading to minimum ε_f) if the data rate is successively reduced. The figure also gives an example that for some situations FEC will not achieve the quality as encoding without FEC (which is quite untypical). In addition, the figure depicts also that FEC with well-chosen priorities may be better than FEC without priorities (which is to be expected).

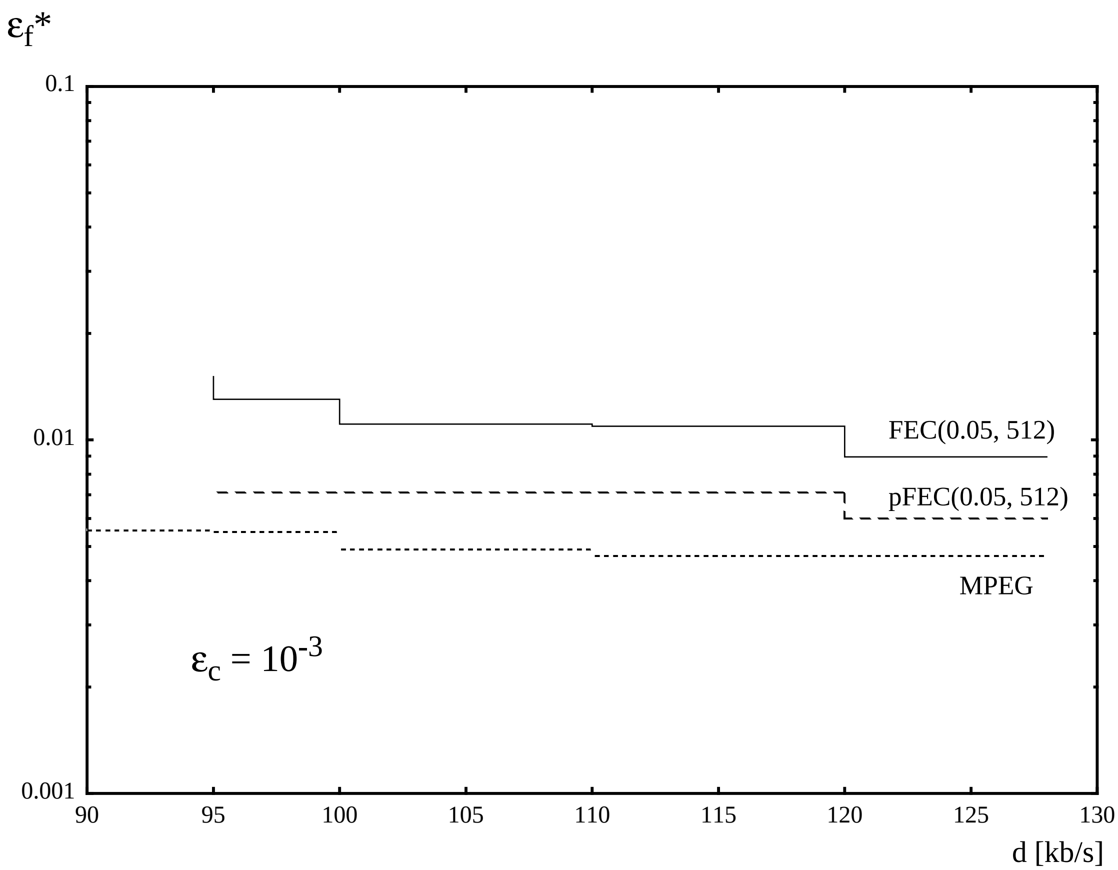

Fig. 5.2: Frame loss probability for and principle usability of selected encoding variants in dependence of the available transmission capacity (fine-grain view)

Table 5.3, for a variety of encoding variants and again for $\varepsilon_c = 0.001$, contains detailed information at which points of throughput reductions the optimum GOP pattern will become unavailable and what effect on $\varepsilon_f{}^*$ this would have (quantification of the increases in $\varepsilon_f{}^*$). It turns out that GOP patterns which become unavailable as a consequence of reductions in available data rate quite often are those patterns which led to optimum ε_f. This means that patterns with a potential of good QoS typically have high throughput requirements (though the reverse is not always true, according to our experimental results).

	128	125	120	115	110
MPEG	0.0047	✓	✓	✓	0.00489
FEC (0.05, 512)	0.00896	✓	0.01094	✓	0.01108
pFEC (0.05, 512)	0.005996	✓	0.007114	✓	✓
FEC (0.05, 128)	4.986×10^{-4}	✓	✓	5.939×10^{-4}	6.965×10^{-4}
pFEC (0.05, 128)	0.01977	✓	✓	0.02201	0.02417
FEC (0.1, 128)	8.603×10^{-8}	✓	✓	✓	✓
pFEC (0.1, 128)	7.688×10^{-6}	✓	2.959×10^{-4}	3.736×10^{-4}	4.322×10^{-4}
FEC (0.2, 128)	1.905×10^{-14}	✓	✓	✓	—
pFEC (0.2, 128)	2.814×10^{-10}	✓	✓	3.820×10^{-10}	—
FEC (0.3, 128)	1.567×10^{-21}	✓	—	—	—
pFEC (0.3, 128)	4.233×10^{-17}	6.727×10^{-17}	—	—	—

	105	100	95	90	85
MPEG	✓	0.005489	0.005544	0.005588	—
FEC (0.05, 512)	✓	0.01304	0.01509	—	—
pFEC (0.05, 512)	✓	✓	0.007203	—	—
FEC (0.05, 128)	8.063×10^{-4}	9.825×10^{-4}	—	—	—
pFEC (0.05, 128)	0.02628	0.02923	—	—	—
FEC (0.1, 128)	✓	—	—	—	—
pFEC (0.1, 128)	4.943×10^{-4}	—	—	—	—
FEC (0.2, 128)	—	—	—	—	—
pFEC (0.2, 128)	—	—	—	—	—
FEC (0.3, 128)	—	—	—	—	—
pFEC (0.3, 128)	—	—	—	—	—

Table 5.3 : Quantification of frame loss probabilities for different encoding variants in dependence of the available transmission capacity; *notation:* ✓: still same value as in column to the left; —: no GOP pattern available for this value of r

Overall, case study I provided us with several interesting insights, namely:

- The elimination of GOP patterns by increasing redundancy for FEC seems to be less “costly” in terms of QoS than the potential gain of the ameliorated FEC (as long as there are still patterns left for encoding).

- For fixed data rate nearly always just one or two GOP patterns lead to the optimum value of ε_f (for various values of ε_c , namely $\varepsilon_c \in \{10^{-10}, 10^{-9}, ..., 10^{-1}\}$). If this would be confirmed by other case studies, it would be extremely favourable, as it would allow one to choose the optimum GOP pattern for a given data rate independently of the packet loss rate (but, of course, dependent on the data rate available).

- There is some hint that the packet size chosen plays a critical role with regard to the video quality achievable using FEC (we will investigate this in-depth in case study IIIa/ section 7).

- Adequate prioritization of frames seems to be critical otherwise FEC without priorities is likely to provide better results.

6. Case Study II: Implications of Varying Network Reliability

In *case study II* our aim has been to investigate the impact of varying the reliability of the network, i.e. the fragment (here packet) loss probability ε_c . During case study II the video stream observed was fixed again (same stream and therefore same frame sizes as in case study I). Moreover, the boundary conditions for case study II as what concerns the communication network and the VC assumed were identical to study I (d = 128 kb/s, $p_l \leq 1024$ Byte, 10 Byte control information per packet).

We varied throughout case study II:

- the packet loss probability ε_c ,
- the amount of redundancy r spent for forward error control (in particular, we chose: $r \in \{0.0, 0.05, 0.1, 0.2, 0.3\}$),
- the packet length used for transmission; in case of MPEG: 1024 Byte, in case of FEC (with and without priorities): 512 Byte combined with r = 0.05 and in case of FEC without priorities, in addition, 128 Byte combined with all values of r.

The results for the frame loss probability ε_f were taken for the GOP pattern which turned out to be best overall for the corresponding variant of encoding. In particular, the GOP pattern which turned out to be optimum was in the case of (cf. notation for *FEC(r, p_l)* and *pFEC(r, p_l)* as in section 5):

- *MPEG*: I B B B P B B B I . . .;
- *pFEC(0.05, 512)*: I B B B P B B B I . . . ($x_I = x_P = 0.87$, $x_B = 1.0$) ;
- *FEC(0.05, 512)*: I B B B B I . . .;
- *FEC(0.05, 128)*: I B B B B B I . . .;
- *FEC(0.1, 128)*: I B B B B B B B B B I . . .;
- *FEC(0.2, 128)*: I B B B B B B B B B I . . .;
- *FEC(0.3, 128)*: I B B B B B B B B B I . . .

Fig. 6.1 depicts the dependency of ε_f on the packet loss probability ε_c for the encoding variants observed, mainly to achieve comparison between MPEG and encoding with FEC without priorities (for the GOP patterns as listed above).

The results show us, e.g., that:

- for all packet loss probabilities FEC with priorities may be better than without priorities, if priorities are chosen in the right way (cf. *pFEC(0.05, 512)* vs. *FEC(0.05, 512)*);
- even with $\varepsilon_c = 0.01$ we can still achieve surprisingly low ε_f (i.e. rather good QoS) if we just use a sufficient amount of redundancy, e.g. $r \in \{0.2, 0.3\}$, and a small enough packet size; note that the results indicate that typically a large amount of redundancy may be

worth-while although this implies that fewer GOP patterns are then still available for encoding (cf. also more “unfavourable” patterns to be used with r = 0.2 and r = 0.3 and for those values of ε_f turning out to be optimum);

- with the same r it is usually advisable to take rather small packet-sizes for transmission (significantly smaller than the maximum packet-size which would be allowed), although smaller packet-sizes again reduce the number of GOP patterns available as the overhead in control information as well as the error probability per bit transmitted become significantly larger with usage of very small packets.

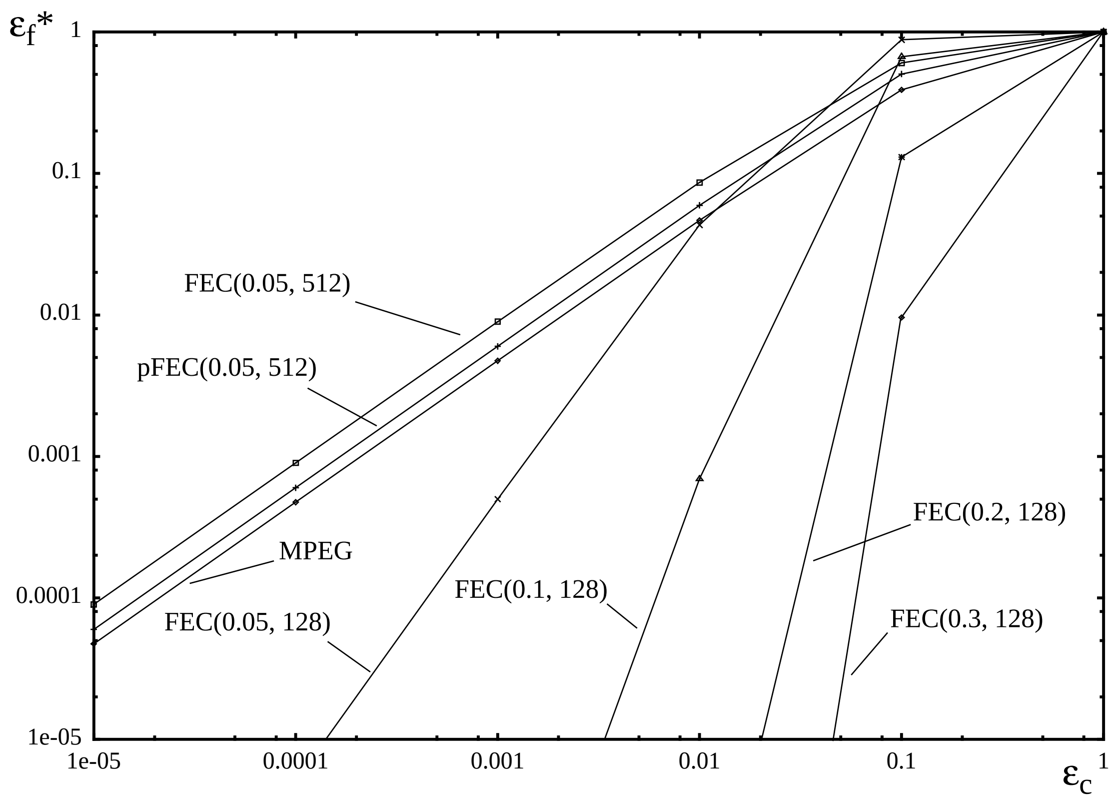

Fig. 6.1: Frame loss probability for and principle usability of different encoding variants in dependence of the packet loss probability

7. Case Study III: Adequate Parametrization for Transmission of Video Streams

Case studies I and II, among others, suggested two kinds of optimization problems in video communication to be investigated in more depth.

First, the results presented up to now make clear that it is very important to determine an adequate size for the packets, which are used for transmitting the data of a video stream (including the possibly added redundancy if FEC is applied). Secondly, the case studies of sections 5 and 6 made evident that a very critical decision in the context of FEC is how much redundancy to spend for FEC in order to achieve the best QoS possible.

C*ase study III* is conceived to solve these two optimization problems, namely study IIIa refers to the study with respect to choice of (close to) optimum packet-size whereas study IIIb investigates the impact on QoS of a varying amount of redundancy in FEC. Evidently, we have to keep in mind that adequate parametrization with regard to packet-size and to the amount of redundancy used, are not completely independent of each other.

C*ase study IIIa* (Determination of packet-size):

The advantage of using small packets is that a packet loss then would have a less negative impact as less data would be lost than with larger packets. There are two disadvantages in using small packets, however, namely that the overhead in control information per packet as well as the loss probability per bit transmitted is getting higher. In addition, fragmentation will not work very well with those packet-sizes which are just slightly smaller than typical video frame sizes. So, overall, it becomes unclear which factor will dominate, if we calculate and compare the QoS achieved by using different packet-sizes. Case study IIIa, again, assumes the same boundary conditions concerning the video stream and the communication network (in particular the VC used) observed as in case study II.

We varied throughout case study IIIa:

- the packet loss probability ε_c, namely $\varepsilon_c \in \{0.1, 0.01, 0.001\}$,
- the amount of redundancy r spent for FEC, in particular: $r \in \{0.0, 0.05, 0.1, 0.2\}$,
- the packet length p_l used for transmission, now varied at a finer granularity than in case study II, namely $p_l \in \{64, 128, 256, 400, 512, 700, 850, 1024 \text{ Byte}\}$.

Fig. 7.1 illustrates results, which, for different variants of encoding, different packet loss probability and different amount of redundancy, show us what packet-sizes would yield minimum $\varepsilon_f{}^*$ (i.e. the best QoS). Although we see that the various factors of influence lead to a situation with several local optima (instead of a single global optimum, i.e. maximum resp. minimum), there exists the general tendency that small packet-sizes seem favourable independent of the value of ε_c (at least as long as: $\varepsilon_c \leq 0.01$).

Fig. 7.1 provides us with some interesting insights: encoding without FEC (i.e. MPEG only) will lead to unacceptable quality already with $\varepsilon_c = 0.01$ (which implies: $\varepsilon_f{}^* = 0.046$). Note that there is no reason to use smaller than maximum size packets in case of MPEG encoding (without FEC). We also realize that encoding with r = 0.2 and $\varepsilon_c = 0.01$ yields better results for all packet-sizes than r = 0.05 and $\varepsilon_c = 0.001$, i.e. use of sufficient redundancy may even compensate for the negative effect of a packet loss probability assumed to be a factor of 10 higher. For r = 0.2, $\varepsilon_c = 0.01$ and $p_l = 64$ Byte the use of FEC will still provide us with very good quality.

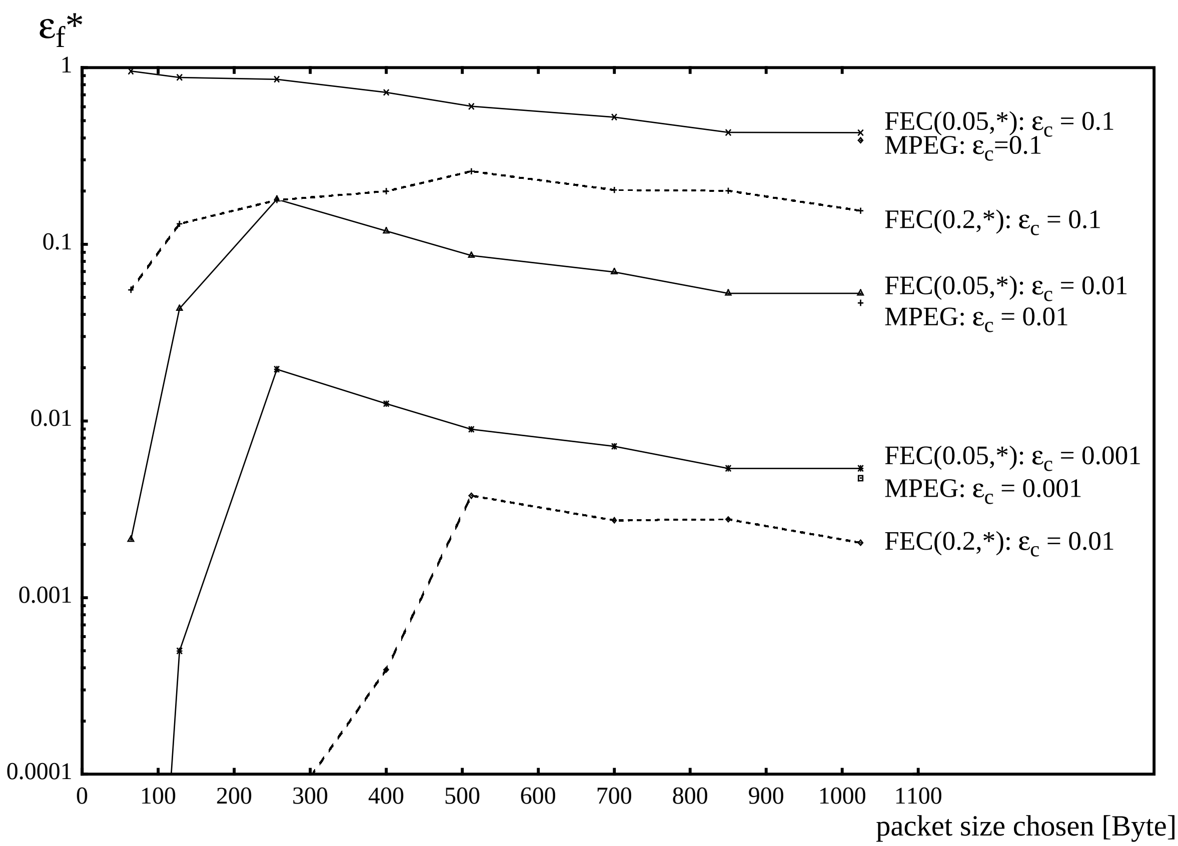

Fig. 7.1: Frame loss probability in dependence of the packet-size chosen for selected encoding variants

C*ase study IIIb* (Determination of amount of redundancy to be spent):

It has already been observed in earlier case studies that, the more redundancy an FEC based encoding algorithm will use, the better the QoS to be expected is. We now want to quantify this observation under the boundary conditions of study IIIa.

We varied throughout case study IIIb:

- the packet loss probability ε_c, namely $\varepsilon_c \in \{0.1, 0.01, 0.001, 0.0001\}$,
- the amount of redundancy r (no priorities in FEC in this study!) spent, in particular: $r \in \{0.0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35\}$, such that, in addition $x_I = x_P = x_B = 1/(1+r)$ as suggested by [Sto 95].

The packet length p_l used for transmission was kept constant now, in particular, $p_l = 1024$ Byte with MPEG and $p_l = 128$ Byte in the case of FEC.

Fig. 7.2 shows that, as expected based on the earlier results, use of as much redundancy as is possible would be advisable in FEC (as long as still GOP patterns exist at all for encoding). Moreover, the advantage of additional redundancy becomes increasingly significant for increasing values of ε_c. The figure also indicates that no GOP pattern would be left for encoding if more than 40% redundancy would be used for FEC (which limits r to $r \leq 0.35$). It should be noted, too, that under the model assumptions made, a sufficiently large r is able to achieve a frame loss probability of $\varepsilon_f^* < 10^{-10}$ even for $\varepsilon_c = 10^{-2}$.

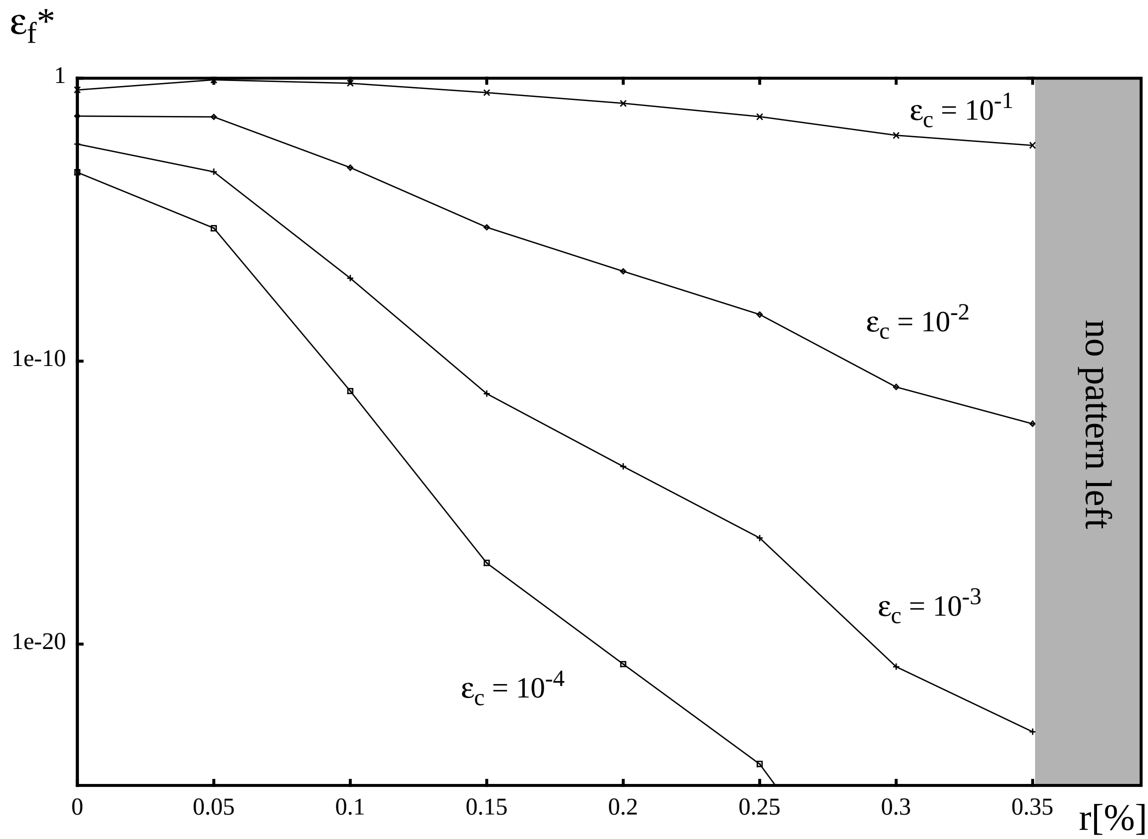

Fig. 7.2: Frame loss probability in dependence of percentage of redundancy used for FEC

8. Comparison of QoS for Different Video Encoding Schemes and QoS Management Based on Analytical QoS Evaluations

To summarize some of the main results of our case studies, let us give examples to demonstrate the potential gain in QoS which may be achievable for different encoding variants. The examples we selected have been part of the studies described in detail in sections 5-7.

These examples illustrate among others that situations exist where

- FEC with priorities is able to provide better QoS than FEC without priorities;
- FEC with priorities will not achieve the same QoS as FEC without priorities, if priorities are not chosen adequately;
- already the change of the packet-size used for transmission may have the impact that an encoding scheme which was superior in QoS for the original packet-size becomes inferior for the new packet-size;
- an encoding variant may be superior to another one for most (or even all) values of ε_c (i.e. independently of ε_c);
- for sufficiently small values of ε_c , the gain in using one coding variant instead of another may easily become several orders of magnitude (w.r.t. $\varepsilon_f{}^*$).

In the examples depicted by Fig. 8.1 the packet/ cell loss probability ε_c has been varied such that $\varepsilon_c \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1.0\}$.

Fig. 8.1 contains the following five comparisons of encoding variants (by way of example):

- C1: boundary conditions as in section 5; comparison of $\varepsilon_f{}^*$ between PET and MPEG; *coding variants*:

  $C_{1,a}$: MPEG; $p_l = 1024$;

  $C_{1,b}$: PET; $p_l = 128$; $r = 0.1$; $x_i = 0.79$; $x_P = 0.86$; $x_B = 0.95$;

  $r_{PM}(C_1; \varepsilon_c) := \varepsilon_f{}^*(C_{1,a}; \varepsilon_c) / \varepsilon_f{}^*(C_{1,b}; \varepsilon_c)$ for all ε_c , i.e. for each value of ε_c we consider the relationship between $\varepsilon_f{}^*$ for coding variant $C_{1,a}$ and $\varepsilon_f{}^*$ for $C_{1,b}$.

- C2: boundary conditions as in section 5; comparison of $\varepsilon_f{}^*$ between PET and MPEG; *coding variants*:

  $C_{2,a}$: MPEG; $p_l = 1024$ (identical to $C_{1,a}$);

  $C_{2,b}$: PET; $p_l = 128$; $r = 0.3$; $x_i = 0.68$; $x_P = 0.7$; $x_B = 0.81$;

  $r_{PM}(C_2; \varepsilon_c) := \varepsilon_f{}^*(C_{2,a}; \varepsilon_c) / \varepsilon_f{}^*(C_{2,b}; \varepsilon_c)$ for all ε_c

- C3: boundary conditions as in section 5 but modifications in VC and video stream considered as follows: ATM based network, i.e. cells transmitted instead of packets (48 Byte payload, 5 Byte Header), T1 link speed (1544 Mb/s); also modified frame lengths namely l_i =

12500 Byte; l_P = 8100 Byte; l_B = 2250 Byte; comparison of $\varepsilon_f{}^*$ between PET and MPEG;
coding variants:
$C_{3,a}$: MPEG; $p_l = 53$;
$C_{3,b}$: PET; $p_l = 53$; $r = 0.1$; $x_i = 0.81$; $x_P = 0.83$; $x_B = 0.96$;
$r_{PM}(C_3; \varepsilon_c) := \varepsilon_f{}^*(C_{3,a}; \varepsilon_c) / \varepsilon_f{}^*(C_{3,b}; \varepsilon_c)$ for all ε_c

- C4: boundary conditions as in section 5; comparison of $\varepsilon_f{}^*$ between FEC without priorities and PET;
  coding variants:
  $C_{4,a}$: FEC without priorities; $p_l = 512$; $r = 0.05$; $x_i = x_P = x_B = 0.95$;
  $C_{4,b}$: PET; $p_l = 512$; $r = 0.05$; $x_i = x_P = 0.87$; $x_B = 1.0$;
  $r_{PF}(C_4; \varepsilon_c) := \varepsilon_f{}^*(C_{4,a}; \varepsilon_c) / \varepsilon_f{}^*(C_{4,b}; \varepsilon_c)$ for all ε_c

- C5: boundary conditions as in section 5; comparison of $\varepsilon_f{}^*$ between FEC without priorities and PET;
  coding variants:
  $C_{5,a}$: PET; $p_l = 128$; $r = 0.05$; $x_i = x_P = 0.87$; $x_B = 1.0$; (inadequate prioritization)
  $C_{5,b}$: FEC without priorities; $p_l = 128$; $r = 0.05$; $x_i = x_P = x_B = 0.95$;
  $r_{FP}(C_5; \varepsilon_c) := \varepsilon_f{}^*(C_{5,a}; \varepsilon_c) / \varepsilon_f{}^*(C_{5,b}; \varepsilon_c)$ for all ε_c

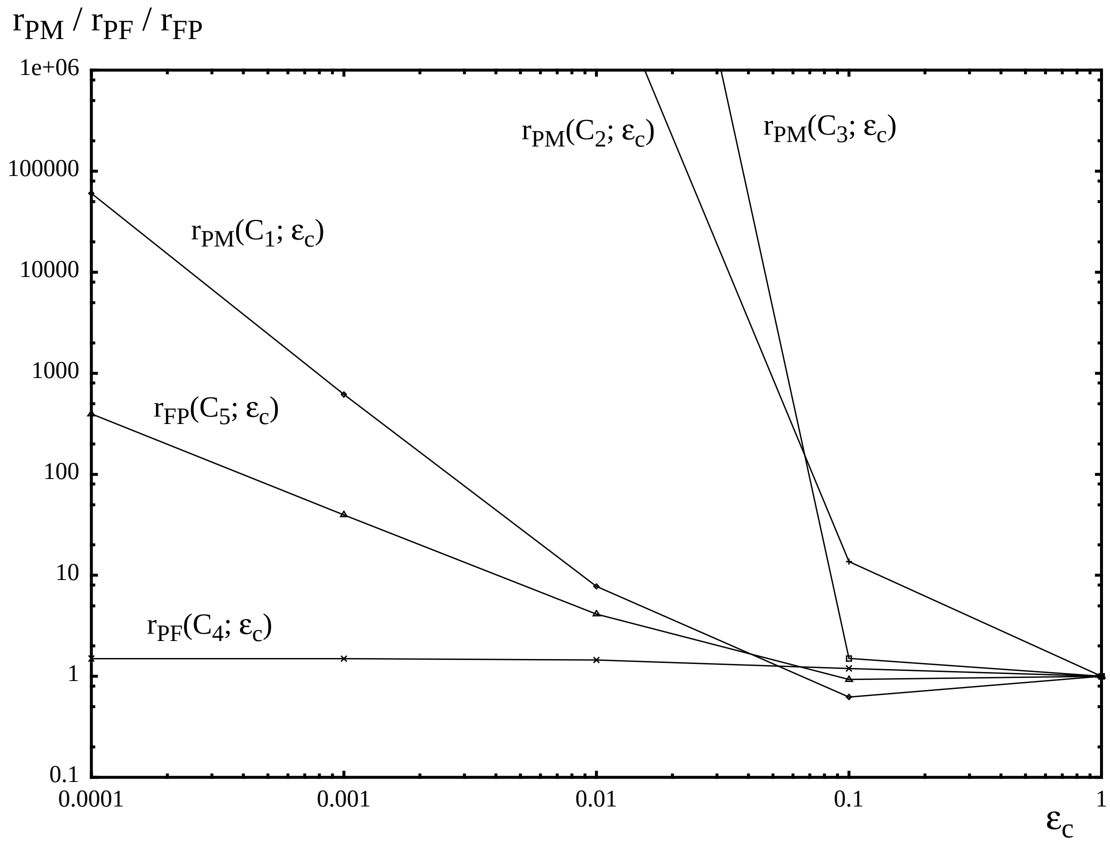

Fig. 8.1: Comparisons between frame loss probabilities as achieved by different encoding schemes (in dependence of cell/ packet loss probability)

Our case studies also indicated various ways to achieve QoS management based on (analytical) models. We want to refine this shortly in the following.

Applying the kind of analytical models we presented in this paper, we could e.g. measure in an existing network and/ or in an endsystem:

- the presently observed cell/ packet loss probability for a VC;
- the distribution of frame lengths resulting from an MPEG encoding applying the presently used GOP pattern for the encoding (thus observing the intensity of motion in the video sequence to be transmitted).

Based on such measurements a component responsible for QoS management ("QoS-Manager") could e.g. take decisions to determine the presently optimum

- GOP pattern for encoding of the video stream to be transmitted;
- packet-size to be favourably used if encoding with FEC is applied;
- amount of redundancy to be used in the case of encoding with FEC.

As the above optimizations (besides determining the optimum amount of redundancy) are typically done only within a relatively small set of possible variants, they can be done by just evaluating all of the variants applying our analytical models. In addition, the QoS-Manager could react dynamically onto changing boundary conditions. As we suggest to apply analytical models for QoS management the reactions could be highly adaptive (e.g. in the order of a few seconds like in routing decisions in communication networks). To conclude let us shortly summarize how model based QoS management by means of our proposed models could look like by indicating the potential activities of a QoS-Manager:

QoS-Manager

- STEP 1:
  Apply measurement results to decide which GOP patterns are possible (taken into account the throughput limitation of VC).
- *STEP 2:* For each GOP pattern do:
  STEP 2a: Calculate probability of losing I-, P- or B-frames (as a consequence of cell/ packet loss probability as it is presently observed for VC).
  STEP 2b: Calculate expected number of lost frames (of any type) within a GOP.
  STEP 2c: Calculate frame loss probability for total video stream.
  STEP 2d: Store P* -- the optimum GOP pattern determined up to now.
- STEP 3:
  Encode according to pattern P* as long as intensity of motion does not vary significantly for given video stream (criterion for changing GOP pattern, e.g.: throughput threshold reached).

9. Summary and Outlook

In this paper we tried to quantitatively assess QoS in video communication from the users' point of view by means of analytical models. In modeling we concentrated on network losses and transmission errors respectively. Several simplifying assumptions were required to obtain models which are still analytically tractable and which thus could be used to support model based QoS management.

A number of case studies with our models gave us several interesting insights in regard to the expected efficiency of different encoding schemes for video communication and their potential to build fault-tolerant distributed applications. Some general observations we made up to now include:

- Encoding with FEC may provide a frame loss probability which is orders of magnitude better than encoding without FEC (even with the same throughput available).
- Optimum values for frame loss probability ($\varepsilon_f{}^*$) are typically achieved with GOP patterns which in turn have relatively high "costs" in throughput requirement.
- A GOP pattern quite often is optimum not only for one special value of packet/ cell loss probability (ε_c) but for the whole range of ε_c-values considered, e.g. $10^{-10} \le \varepsilon_c \le 1$.
- With FEC and packet-switching networks it seems to be favourable not to use the maximum packet-size but a smaller value (although smaller packets imply more overhead in transmitting control information).
- Quite often it is unfavourable to use strongly different priorities for different frame types with FEC; as is to be expected, it turns out to be better not to use any priorities than priorities which are inadequately chosen.
- One should not try to minimize the redundancy spent for FEC, quite often additional redundancy yields significantly better QoS (evidently, as long as the overall throughput limitation is respected).
- Under our assumptions, when using FEC with large amount of redundancy, sufficiently good QoS can be achieved even in very "lossy" network environments as might be the case in wireless communication systems.

In interpreting the results of our case studies one has to keep in mind our central assumption of mutually indepent packet/ cell losses for VCs. Thus, our models in particular do not reflect situations where losses or transmission errors occur in bursts. Simulation models would allow one to cover such situations. Calculating ε_f based on mean frame lengths only represents an approximation, too. However, it is easy to obtain bounds for ε_f by using our models (just replacing mean lengths by maximum resp. minimum frame lengths).

Additional case studies are still planned by us, in particular to look at:

- additional video sequences (e.g. with different intensity of motion in the scenes);
- more examples which are assuming ATM-based networks;

- wireless network configurations, which could be good examples of networks reflected quite realistically by our models.

Further investigations are considered by us in order to determine additional boundary conditions of networks and video streams to be transmitted, where selected redundancy, i.e. prioritization of frames, really implies significant gain in QoS (when using algorithms based on FEC). Moreover, we plan to apply slightly modified models covering FEC and prioritization which would reflect a more efficient usage of PET-like algorithms, where transport of I-/P-/B-frame specific data (including the corresponding redundancy) would be organized in such a way that fragments are used which are always responsible for transporting data of only one type of frame.

We hope that these case studies together with additional experiments to continue model validation, will show that, even with analytical models, interesting and sufficiently valid insight can be gained into the QoS as achieved by different video encoding schemes.

Acknowledgements

The author would like to thank Dr. Andres Albanese for his initial stimulation to evaluate the performance of video encoding schemes and for a lot of in-depth discussions during elaboration of the models introduced in this paper and he is also very grateful to Jan-Peter Richter for some support in the evaluation of case studies. The considerable assistance of Dr. Marco Alfano in realizing the programs to analytically evaluate the models presented is greatly appreciated, too. Finally, words of thank go to Drs. Michael Luby and Rainer Storn for valuable hints with regard to adequate PET modeling and to desirable case studies.

10. References

[ABE 94] A. Albanese, J. Bloemer, J. Edmonds, M. Luby: Priority Encoding Transmission, ICSI report, TR-94-039 (1994), 35 p.

[AlL 96] A. Albanese, M. Luby: PET -- Priority Encoding Transmission, in: W. Effelsberg et al. (eds.): *High-Speed Networking for Multimedia Applications,* Kluwer Academic Publ. (1996), 247 - 265

[Bai 96] G. Bai: Measurements for MPEG Encoded Video Sequences, (private communication)

[DeP 96] M. De Prycker: Asynchronous Transfer Mode, *Prentice Hall*, 1996

[DeM 95] H. de Meer: Modeling and Management of Responsive Systems: The Quality of Service for Multi-Media Applications in High Speed Networks as an Example, *2nd ISSAT Int. Conf. on Reliability and Quality in Design*, Orlando (1995)

[EcS 96] D. Eckhardt, P. Steenkiste: Measurement and Analysis of the Error Characteristics of an In-Building Wireless Network, *ACM SIGCOMM'96 Conf.,* Computer Communication Review, Vol. 26, No. 4 (1996), 243 - 254

[ILD 94] M.R. Ismail, I.E. Lambadaris, M. Devetsikiotis, A.R. Kaye: Frame Spreading and a Modified Priority Scheme for VBR MPEG Video Transmission in ATM Networks, *Canadian Conf. on Electrical and Computer Engineering*, Halifax (1994)

[ILD 95] M.R. Ismail, I.E. Lambadaris, M. Devetsikiotis, A.R. Kaye: Modeling Prioritized MPEG Video Using TES and a Frame Spreading Strategy for Transmission in ATM Networks, *INFOCOM'95* (1995)

[KMS 95] D. Kozen, Y. Minsky, B. Smith: Efficient Algorithms for Optimal Video Transmission, Report Computer Science Department, Cornell University (1995)

[LeG 91] D. Le Gall: MPEG: A Video Compression Standard for Multimedia Applications, *C.ACM*, Vol. 34, No. 4 (1991), 30 - 44

[RiR 95] M.J. Riley, I.E.G. Richardson: FEC and Multi-layer Video Coding for ATM Networks, in: *Performance Modelling and Evaluation of ATM Networks*, Vol. 1, Chapman & Hall (1995), 450 - 457

[Sto 95] R. Storn: Modeling and Optimization of Pet-Redundancy Assignment for MPEG Sequences, ICSI report, TR-95-018 (1995), 24 p.

[Tan 96] A. Tanenbaum: Computer Networks, 3rd edition, *Prentice-Hall* (1996)

[WoB 95] B. Wolfinger, G. Bai: Lastmessungen fuer Videoverkehrsquellen als Basis fuer eine realitaetsnahe Modellierung von Echtzeit-Kommunikationssystemen, in: P. Holleczek (ed.): *PEARL'95*, Springer-Verlag (1995) (in German), 81 - 91

[WoK 90] B. Wolfinger, J.J. Kim: Load Measurements as a Basis for Modeling the Load of Innovative Communication Systems with Service Integration, Proc. 2nd IEEE Workshop on *Future Trends of Distributed Computing Systems*, Kairo (1990)

[ZhF 96] H. Zhu, V.S. Frost: In-Service Monitoring for Cell Loss Quality of Service Violations in ATM Networks, *IEEE / ACM Transactions on Networking*, Vol. 4, No. 2 (1996), 240 - 248