On the Potential of FEC Algorithms in Building Fault-tolerant Distributed Applications to Support High QoS Video Communications

Bernd E. Wolfinger [1)]

Hamburg University, Computer Science Department,
Telecommunications and Computer Networks Division

Abstract

Studies for distributed applications have demonstrated that application quality in distributed computing is strongly dependent on the quality of the underlying communication system. Besides performance characteristics the reliability properties of a communication system are playing a predominant role with regard to the quality achievable from applications' point of view.

In this paper we study distributed applications with video communication via lossy communication systems. We develop analytical models reflecting transmission of MPEG coded video streams both without and with forward error correction (FEC). These models allow us to determine the optimum amount of redundancy to be used for FEC when communicating via constant bit rate connections. We conclude that FEC algorithms offer the potential for high quality-of-service even in the case of using a communication system with low quality.

1. Motivation

A lot of earlier studies for distributed applications have demonstrated that application quality in distributed computing (as in parallel computing) is strongly dependent on the quality of the underlying communication system or communication network that is used for data exchange between application processes. Besides performance characteristics (such as throughput or transmission delays) the reliability properties of a communication system (such as probability of losing or corrupting data during transmission) are playing a predominant role with regard to the quality achievable when realizing a distributed application. In distributed computations, in particular, performance typically represents the prime quality criterion whereas in distributed applications with real-time requirements (such as real-time audio/video communication) reliability of the underlying network is prevailing because timing restrictions usually do not allow for retransmissions in case of transmission errors or losses.

In this contribution, we want to focus on distributed applications with video communication (such as video conferencing or video-on-demand [Ste 94]) as an important class of innovative distributed applications. As with other distributed applications video-oriented applications, too, are strongly dependent on communication network quality [NPS 94] and, as in the case of other applications, a network offering its services with a high quality-of-service (QoS) considerably facilitates the building-up of highly efficient and/or reliable applications[StN 95]. In this paper, however, we want to demonstrate that in distributed video-oriented applications a high quality is achievable on Application Layer even in the case that only a communication network offering a low QoS is available. This observation is particularly important as in many situations a low QoS communication network is the only network available when building-up a distributed application. A high QoS network may be unavailable, e.g., because of being non-existent at all (e.g. with wireless communication) or being too expensive (e.g. with wide-area networks / WANs).

In this paper, we assume communication networks, which offer low QoS with respect to their reliability, in particular, we assume highly lossy networks (note that unrecoverable transmission errors are also considered as data losses on Application Layer). Our aim is to show how forward error control (FEC) on Application Layer allows one to build-up fault-tolerant distributed applications with high QoS video communication. We call these applications "smart applications". The errors, most of which can be tolerated by such smart applications are the unrecoverable transmission errors and data losses within the communication network. Our main goal is to develop valid models, which allow us to quantitatively evaluate FEC algorithms (with and without prioritization of data transmitted) with respect to their ability to "mask" data losses of the network on Application Layer by means of transmitting redundancy. The models presented by us reflect transmission of MPEG coded video streams [LeG 91] and they are still analytically tractable. We assume that FEC is achieved using PET algorithm [ABE 94]. The paper comprises some rather comprehensive case studies applying our models. These series of experiments demonstrate impressively how significant even a very low QoS (related to data losses) offered by a communication network may be ameliorated by an adequately chosen FEC on Application Layer. However, our results also show that an appropriate parametrization of the FEC algorithm used is extremely critical for obtaining high QoS for end-users. This parametrization comprises a careful tuning of FEC parameters to the actual state of the network and to the transmission capacity required dynamically by the application. To achieve such a tuning, the models presented in the paper may be directly used, too, which is also demonstrated by us by means of additional case studies.

[1)] research at International Computer Science Institute (ICSI), Berkeley, USA during sabbatical leave from Hamburg University

E-mail: wolfinger@informatik.uni-hamburg.de

2. Introduction

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] 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 [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]. 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).

Let us now shortly describe 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, 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.

3. 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.

We tried to choose model assumptions for our analytical models, which are still sufficiently realistic in order to make the resulting models applicable for a relatively large set 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).

One should note 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.

To conclude this section let us summarize our model assumed for video communication, before attacking the problem of analytical model evaluation in sections 4 and 5. 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 4 covers MPEG encoding only (i.e. no FEC) and section 5 covers MPEG encoding combined with FEC (both with and without priorities for transmitted data units).

4. 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:

- (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: $x_i \leq x_P \leq x_B$;
  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 or maximum packet size respectively, (in [Byte]).

Video Stream:

- $l_i(N,M)$, $l_P(N,M)$, $l_B(N,M)$: 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 or during frame transmission respectively
- ε_f: loss probability reflecting direct and indirect losses of any frames of a total video stream (as defined in section 3)
- 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 applied****)*

- **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 $\varepsilon_f = \varepsilon_f(M_c)$. Evidently, $\varepsilon_f = n_\varepsilon / N$.
- **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 2 and 3.

In Appendix A the interested reader finds a closed-form solution for the analytical model, covering MPEG encoding, and a refinement of the principle steps 2a and 2b of our basic algorithm (for more details cf. [Wol 97a] and [Wol 97b]).

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

Let the notation as introduced in section 4 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 $\varepsilon_f = \varepsilon_f(M_c) = n_\varepsilon / N$.
- **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 4, we want to refine the principle steps 2a and 2b of our basic algorithm (again, more details are given in [Wol 97a] and [Wol 97b]).

- **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

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

where $\lceil x \rceil$ denotes the “ceiling function”.

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”;

(analogous results for z_P and z_B).

Straight-forward argumentation yields [Wol 97a], [Wol 97b]:

$$n_\varepsilon = N \times f(\varepsilon_c, z_i, n_c) +$$
$$(N-1) \times f(\varepsilon_c, z_P, z_i - 1) \times \delta_P +$$
$$(N - N/M) \times f(\varepsilon_c, z_B, z_{P/i} - 1) \times \delta_B +$$
$$(M-1) \times f(\varepsilon_c, z_i, n_c) \times \left(1 - f(\varepsilon_c, z_B, n_c)\right) \times \delta_B$$

(eq.),*

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

$$f(p, m, n) = \begin{cases} 0, \text{ if } (p = 0) \text{ OR } (m > n) \text{ OR } ((p = 1) \text{ AND } (n \neq n_c)) \\ 1, \text{ if } (p = 1) \text{ AND } (n = n_c) \\ \sum_{j=m}^{n} \binom{n_c}{j} \times p^j \times (1-p)^{n_c - j}, \text{ otherwise} \end{cases}$$

FEC without priorities

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

$$n_\varepsilon = N \times f(\varepsilon_c, z, n_c) +$$
$$(M-1) \times f(\varepsilon_c, z, n_c) \times \left(1 - f(\varepsilon_c, z, n_c)\right) \times \delta_B$$

as now $z_i = z_P = z_B = \text{const} = z$.

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".

6. 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).
- 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; cf. similar measurement results for MPEG encoded video streams in [WoB 95].

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 $p_l \in \{128, 512\}$); *FEC(r, *)* if 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: $x_i = 0.68$, $x_P = 0.7$, $x_B = 0.81$.

Fig. 6.1 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 of this first case study 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. 6.1 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 a fixed amount of redundancy).

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 $\varepsilon_c \in \{10^{-10}, 10^{-9}, ..., 10^{-1}\}$). This is 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.
- Adequate prioritization of frames seems to be critical otherwise FEC without priorities is likely to provide better results.

7. 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.

Fig. 7.1 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 those GOP patterns which yield to optimum QoS, i.e. optimum ε_f).

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.

8. 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 6 and 7 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. 8.1 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. 8.1 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.

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. 8.2 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 $\varepsilon_c = 10^{-2}$.

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.

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 during experimentation 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. $10^{-10} \leq \varepsilon_c \leq 1$.
- 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.

Our case studies show that, even with analytically tractable models, interesting insight can be gained into the QoS as achieved by different video encoding schemes and that highly fault-tolerant distributed applications can be built using appropriately parametrized FEC algorithms. The later is true even if such "smart applications" have to communicate via a highly unreliable network.

References

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

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

[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

[NPS 94] J.-P. Nussbaumer, B.V. Patel, F. Schaffa et al.: Networking Requirements for Interactive Video on Demand, IBM Research Report RC-19710 (1994)

[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

[Ste 94] R. Steinmetz (ed.): Multimedia: Advanced Teleservices and High-Speed Communication Architectures. *LNCS Vol. 868*, Springer-Verlag (1994)

[StN 95] R. Steinmetz, K. Nahrstedt: Multimedia: Computing, Communications & Applications, *Prentice- Hall,* 1995

[Sto 95] R. Storn: Modeling and Optimization of Pet-Redundancy Assignment for MPEG Sequences, Internat. Computer Science Institute, ICSI report, TR-95-018, Berkeley (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)

[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)

[Wol 97a] B. Wolfinger: Efficiency of PET and MPEG Encoding for Video Streams: Analytical QoS Evaluations, Internat. Computer Science Institute, ICSI report, TR-97-015, Berkeley (1997), 35 p.

[Wol 97b] B. Wolfinger: Efficiency of Video Encoding Schemes without and with FEC: Analytical Models for QoS Evaluations, *1st World Congress on Systems Simulation*, Singapore (1997)

[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

Appendix A : Closed-form solution for the analytical model, covering MPEG encoding

- **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:

Our model has to cover all the implications of direct losses of frames on a given GOP (here GOP_x).

We now discuss the implications of different types of direct frame losses which may induce (one or more) indirect frame losses for GOP_x :

(a) I-frame of GOP_x lost

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) P-frame of GOP_x lost

The expected impact of directly lost P-frames on GOP_x is such that:

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

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

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) B-frame of GOP_x lost

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) such that the overall impact is as follows:

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

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) I-frame of GOP_{x+1} lost

Loss of I-frame of GOP_{x+1} affects just last B-Group of GOP_x. So, the impact is:

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

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).

Appendix B : Results of case studies (I-IIIa,b)

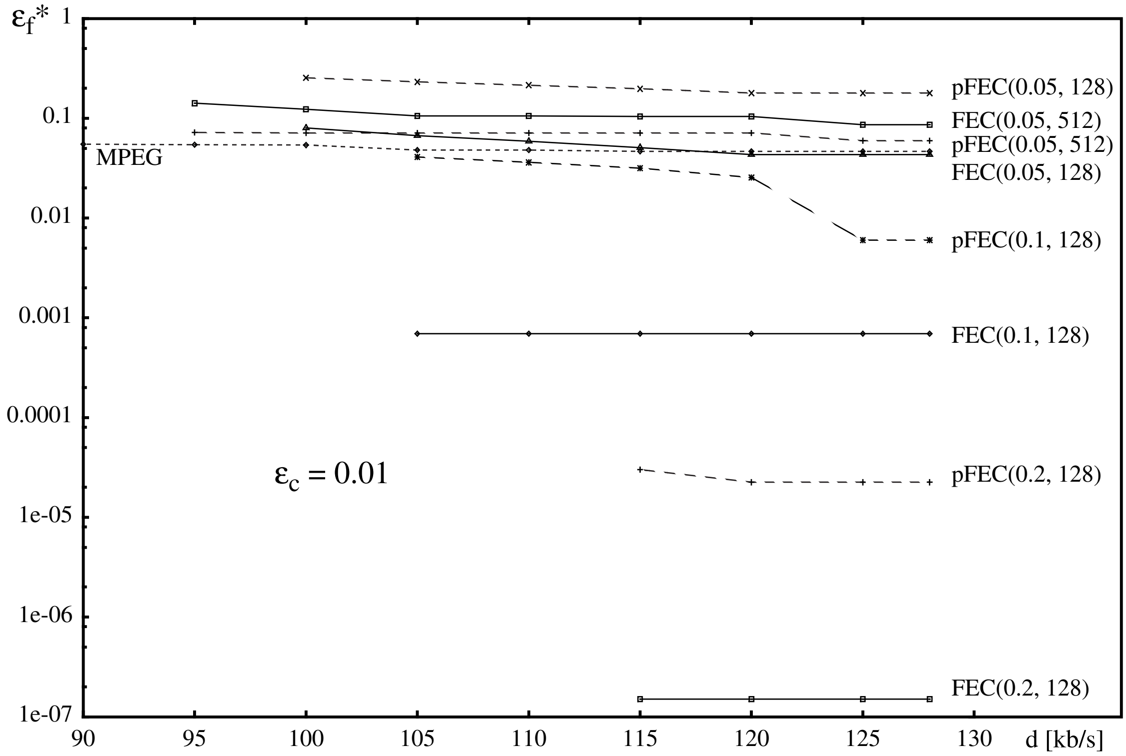

Fig 6.1: Frame loss probability for and principle usability of different encoding variants in dependence of the available transmission capacity (coarse-grain view) (in case of r = 0.3: $\varepsilon_f^* << 10^{-8}$ for all d, thus, those results are not covered by Fig. 6.1)

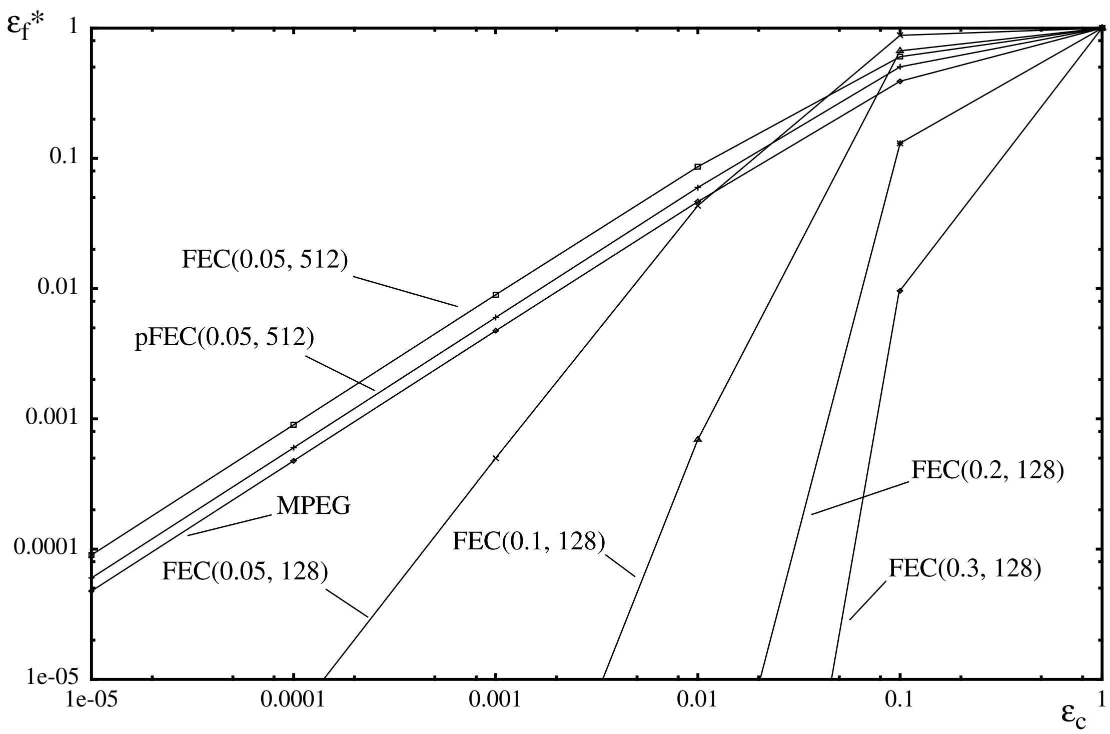

Fig 7.1: Frame loss probability for and principle usability of different encoding variants in dependence of the packet loss probability

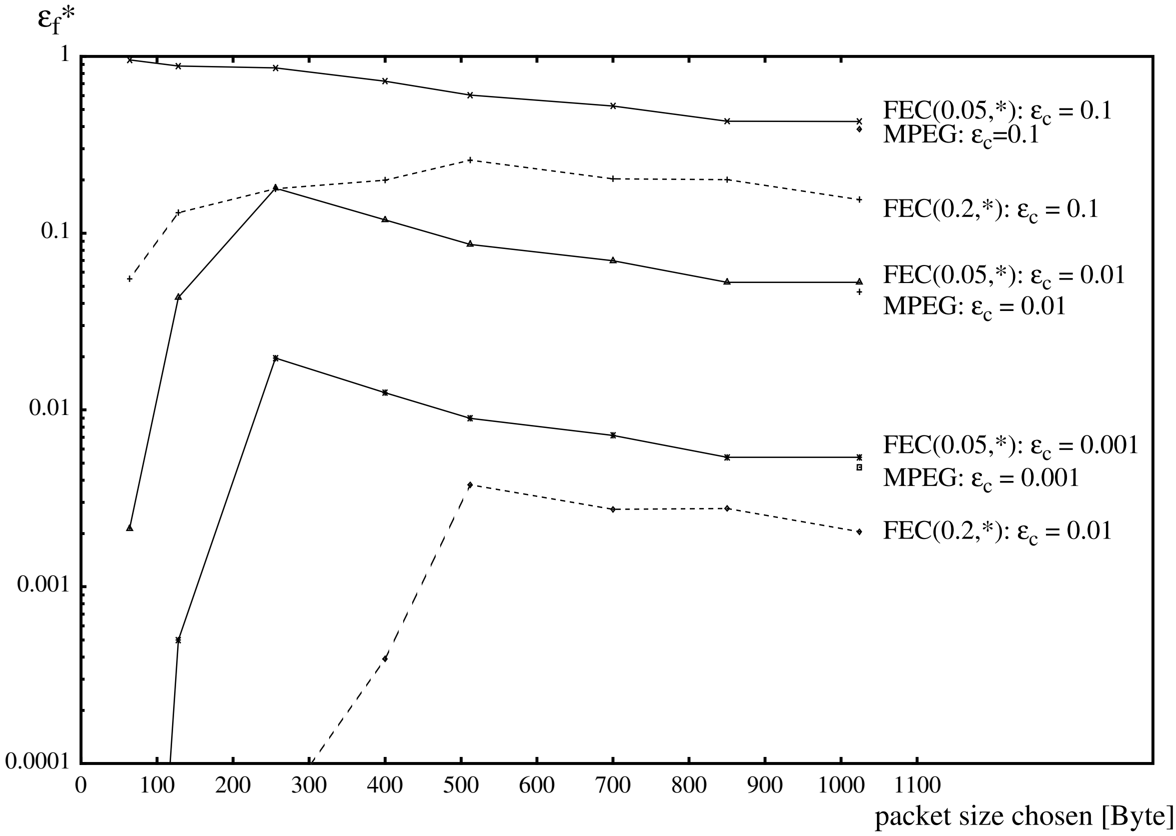

Fig. 8.1: Frame loss probability in dependence of the packet-size chosen for selected encoding variants

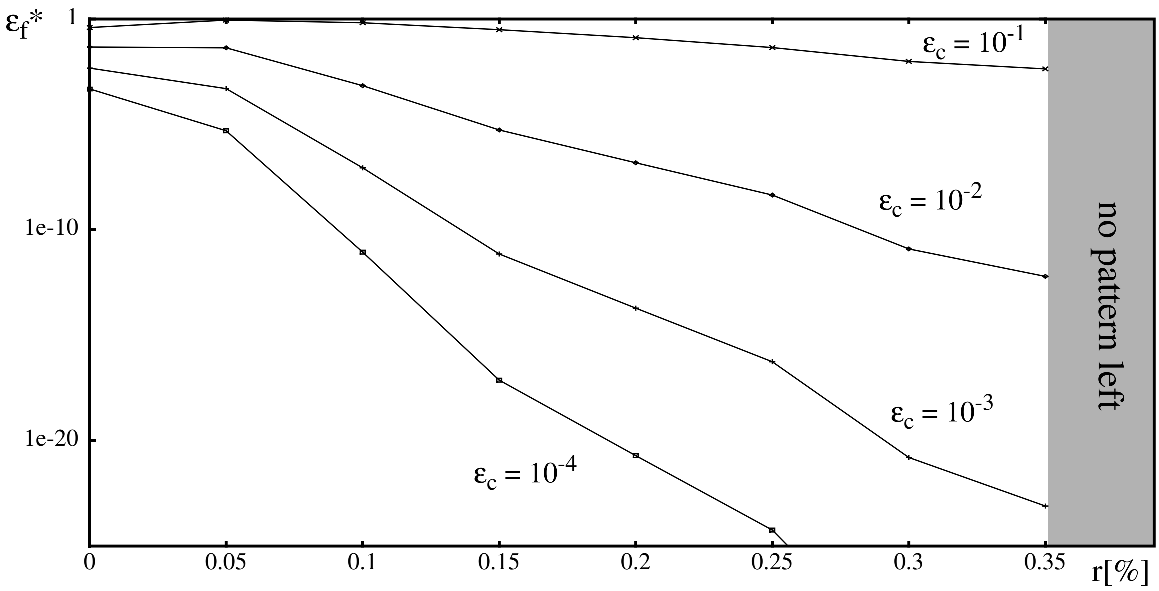

Fig 8.2: Frame loss probability in dependence of percentage of redundancy used for FEC

On the Potential of FEC Algorithms in Building Fault-tolerant Distributed Applications to Support High QoS Video Communications

Bernd E. Wolfinger [1)]

Hamburg University, Computer Science Department,
Telecommunications and Computer Networks Division

Abstract

Studies for distributed applications have demonstrated that application quality in distributed computing is strongly dependent on the quality of the underlying communication system. Besides performance characteristics the reliability properties of a communication system are playing a predominant role with regard to the quality achievable from applications' point of view.

In this paper we study distributed applications with video communication via lossy communication systems. We develop analytical models reflecting transmission of MPEG coded video streams both without and with forward error correction (FEC). These models allow us to determine the optimum amount of redundancy to be used for FEC when communicating via constant bit rate connections. We conclude that FEC algorithms offer the potential for high quality-of-service even in the case of using a communication system with low quality.

1. Motivation

A lot of earlier studies for distributed applications have demonstrated that application quality in distributed computing (as in parallel computing) is strongly dependent on the quality of the underlying communication system or communication network that is used for data exchange between application processes. Besides performance characteristics (such as throughput or transmission delays) the reliability properties of a communication system (such as probability of losing or corrupting data during transmission) are playing a predominant role with regard to the quality achievable when realizing a distributed application. In distributed computations, in particular, performance typically represents the prime quality criterion whereas in distributed applications with real-time requirements (such as real-time audio/video communication) reliability of the underlying network is prevailing because timing restrictions usually do not allow for retransmissions in case of transmission errors or losses.

In this contribution, we want to focus on distributed applications with video communication (such as video conferencing or video-on-demand [Ste 94]) as an important class of innovative distributed applications. As with other distributed applications video-oriented applications, too, are strongly dependent on communication network quality [NPS 94] and, as in the case of other applications, a network offering its services with a high quality-of-service (QoS) considerably facilitates the building-up of highly efficient and/or reliable applications[StN 95]. In this paper, however, we want to demonstrate that in distributed video-oriented applications a high quality is achievable on Application Layer even in the case that only a communication network offering a low QoS is available. This observation is particularly important as in many situations a low QoS communication network is the only network available when building-up a distributed application. A high QoS network may be unavailable, e.g., because of being non-existent at all (e.g. with wireless communication) or being too expensive (e.g. with wide-area networks / WANs).

In this paper, we assume communication networks, which offer low QoS with respect to their reliability, in particular, we assume highly lossy networks (note that unrecoverable transmission errors are also considered as data losses on Application Layer). Our aim is to show how forward error control (FEC) on Application Layer allows one to build-up fault-tolerant distributed applications with high QoS video communication. We call these applications "smart applications". The errors, most of which can be tolerated by such smart applications are the unrecoverable transmission errors and data losses within the communication network. Our main goal is to develop valid models, which allow us to quantitatively evaluate FEC algorithms (with and without prioritization of data transmitted) with respect to their ability to "mask" data losses of the network on Application Layer by means of transmitting redundancy. The models presented by us reflect transmission of MPEG coded video streams [LeG 91] and they are still analytically tractable. We assume that FEC is achieved using PET algorithm [ABE 94]. The paper comprises some rather comprehensive case studies applying our models. These series of experiments demonstrate impressively how significant even a very low QoS (related to data losses) offered by a communication network may be ameliorated by an adequately chosen FEC on Application Layer. However, our results also show that an appropriate parametrization of the FEC algorithm used is extremely critical for obtaining high QoS for end-users. This parametrization comprises a careful tuning of FEC parameters to the actual state of the network and to the transmission capacity required dynamically by the application. To achieve such a tuning, the models presented in the paper may be directly used, too, which is also demonstrated by us by means of additional case studies.

[1)] research at International Computer Science Institute (ICSI), Berkeley, USA during sabbatical leave from Hamburg University

E-mail: wolfinger@informatik.uni-hamburg.de

2. Introduction

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] 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 [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]. 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).

Let us now shortly describe 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, 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.

3. 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.

We tried to choose model assumptions for our analytical models, which are still sufficiently realistic in order to make the resulting models applicable for a relatively large set 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).

One should note 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.

To conclude this section let us summarize our model assumed for video communication, before attacking the problem of analytical model evaluation in sections 4 and 5. 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 4 covers MPEG encoding only (i.e. no FEC) and section 5 covers MPEG encoding combined with FEC (both with and without priorities for transmitted data units).

4. 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:

- (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 or maximum packet size respectively, (in [Byte]).

Video Stream:

- $l_i(N,M)$, $l_P(N,M)$, $l_B(N,M)$: 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 or during frame transmission respectively
- ε_f: loss probability reflecting direct and indirect losses of any frames of a total video stream (as defined in section 3)
- 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 applied**)*

- **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: $|GOP| \le 10$ (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 $\varepsilon_f = \varepsilon_f(M_c)$. Evidently, $\varepsilon_f = n_\varepsilon / N$.
- **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 2 and 3.

In Appendix A the interested reader finds a closed-form solution for the analytical model, covering MPEG encoding, and a refinement of the principle steps 2a and 2b of our basic algorithm (for more details cf. [Wol 97a] and [Wol 97b]).

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

Let the notation as introduced in section 4 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: $|GOP| \le 10$.
- **STEP2:** For every coding pattern $M_c \in M_P$ repeat:
  STEP2a: Determine probability of losing ≥ k fragments within a GOP
  STEP2b: Determine n_ε
  STEP2c: Determine $\varepsilon_f = \varepsilon_f(M_c) = n_\varepsilon / N$.
- **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 4, we want to refine the principle steps 2a and 2b of our basic algorithm (again, more details are given in [Wol 97a] and [Wol 97b]).

- **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

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

where $\lceil x \rceil$ denotes the “ceiling function”.

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

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

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”;

(analogous results for z_P and z_B).

Straight-forward argumentation yields [Wol 97a], [Wol 97b]:

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

(eq.)*,

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

$$f(p, m, n) = \begin{cases} 0, \text{ if } (p = 0) \text{ OR } (m > n) \text{ OR } ((p = 1) \text{ AND } (n \neq n_c)) \\ 1, \text{ if } (p = 1) \text{ AND } (n = n_c) \\ \sum_{j=m}^{n} \binom{n_c}{j} \times p^j \times (1-p)^{n_c - j}, \text{ otherwise} \end{cases}$$

FEC without priorities

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 $z_i = z_P = z_B = \text{const} = z$.

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".

6. 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).
- 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; cf. similar measurement results for MPEG encoded video streams in [WoB 95].

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 $p_l \in \{128, 512\}$); *FEC(r, *)* if 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
  $r = 0.05$: $x_i = x_P = 0.87$, $x_B = 1.0$,
  $r = 0.1$: $x_i = 0.79$, $x_P = 0.86$, $x_B = 0.95$,
  $r = 0.2$: $x_i = 0.71$, $x_P = 0.77$, $x_B = 0.88$,
  $r = 0.3$: $x_i = 0.68$, $x_P = 0.7$, $x_B = 0.81$.

Fig. 6.1 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 of this first case study 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. 6.1 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 a fixed amount of redundancy).

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 $\varepsilon_c \in \{10^{-10}, 10^{-9}, ..., 10^{-1}\}$). This is 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.
- Adequate prioritization of frames seems to be critical otherwise FEC without priorities is likely to provide better results.

7. 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.

Fig. 7.1 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 those GOP patterns which yield to optimum QoS, i.e. optimum ε_f).

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.

8. 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 6 and 7 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. 8.1 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 ε_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. 8.1 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.

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. 8.2 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 $r \le 0.35$). 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 $\varepsilon_c = 10^{-2}$.

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.

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 during experimentation 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.

Our case studies show that, even with analytically tractable models, interesting insight can be gained into the QoS as achieved by different video encoding schemes and that highly fault-tolerant distributed applications can be built using appropriately parametrized FEC algorithms. The later is true even if such "smart applications" have to communicate via a highly unreliable network.

References

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

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

[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

[NPS 94] J.-P. Nussbaumer, B.V. Patel, F. Schaffa et al.: Networking Requirements for Interactive Video on Demand, IBM Research Report RC-19710 (1994)

[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

[Ste 94] R. Steinmetz (ed.): Multimedia: Advanced Teleservices and High-Speed Communication Architectures. *LNCS Vol. 868*, Springer-Verlag (1994)

[StN 95] R. Steinmetz, K. Nahrstedt: Multimedia: Computing, Communications & Applications, *Prentice- Hall,* 1995

[Sto 95] R. Storn: Modeling and Optimization of Pet-Redundancy Assignment for MPEG Sequences, Internat. Computer Science Institute, ICSI report, TR-95-018, Berkeley (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)

[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)

[Wol 97a] B. Wolfinger: Efficiency of PET and MPEG Encoding for Video Streams: Analytical QoS Evaluations, Internat. Computer Science Institute, ICSI report, TR-97-015, Berkeley (1997), 35 p.

[Wol 97b] B. Wolfinger: Efficiency of Video Encoding Schemes without and with FEC: Analytical Models for QoS Evaluations, *1st World Congress on Systems Simulation*, Singapore (1997)

[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

Appendix A : Closed-form solution for the analytical model, covering MPEG encoding

- **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:

Our model has to cover all the implications of direct losses of frames on a given GOP (here GOP_x).

We now discuss the implications of different types of direct frame losses which may induce (one or more) indirect frame losses for GOP_x :

(a) I-frame of GOP_x lost

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) P-frame of GOP_x lost

The expected impact of directly lost P-frames on GOP_x is such that:

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

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

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) B-frame of GOP_x lost

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) such that the overall impact is as follows:

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

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) I-frame of GOP_{x+1} lost

Loss of I-frame of GOP_{x+1} affects just last B-Group of GOP_x. So, the impact is:

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

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).

Appendix B : Results of case studies (I-IIIa,b)

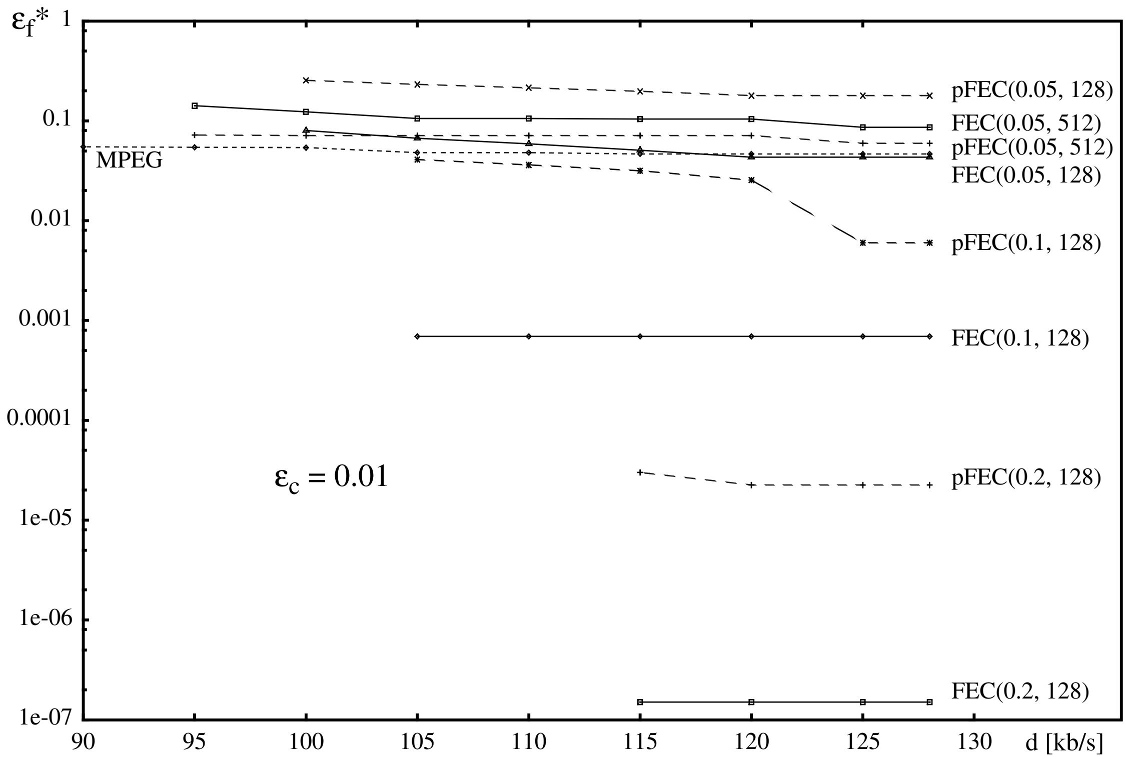

Fig 6.1: Frame loss probability for and principle usability of different encoding variants in dependence of the available transmission capacity (coarse-grain view) (in case of r = 0.3: $\varepsilon_f^* << 10^{-8}$ for all d, thus, those results are not covered by Fig. 6.1)

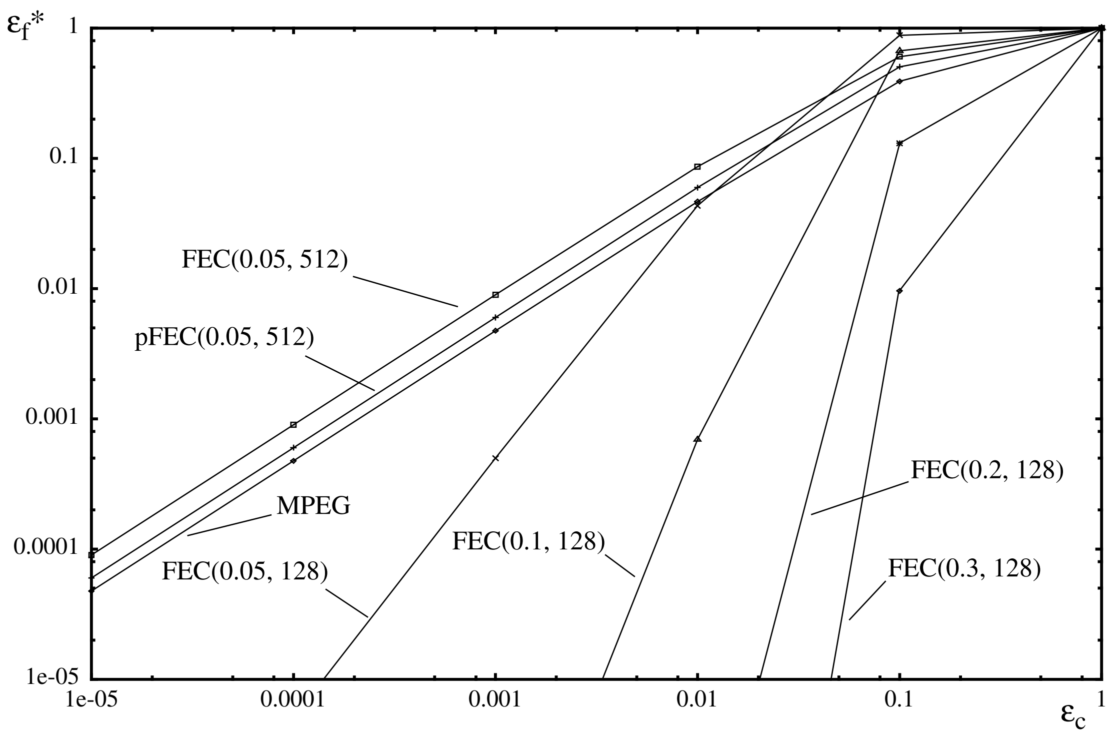

Fig 7.1: Frame loss probability for and principle usability of different encoding variants in dependence of the packet loss probability

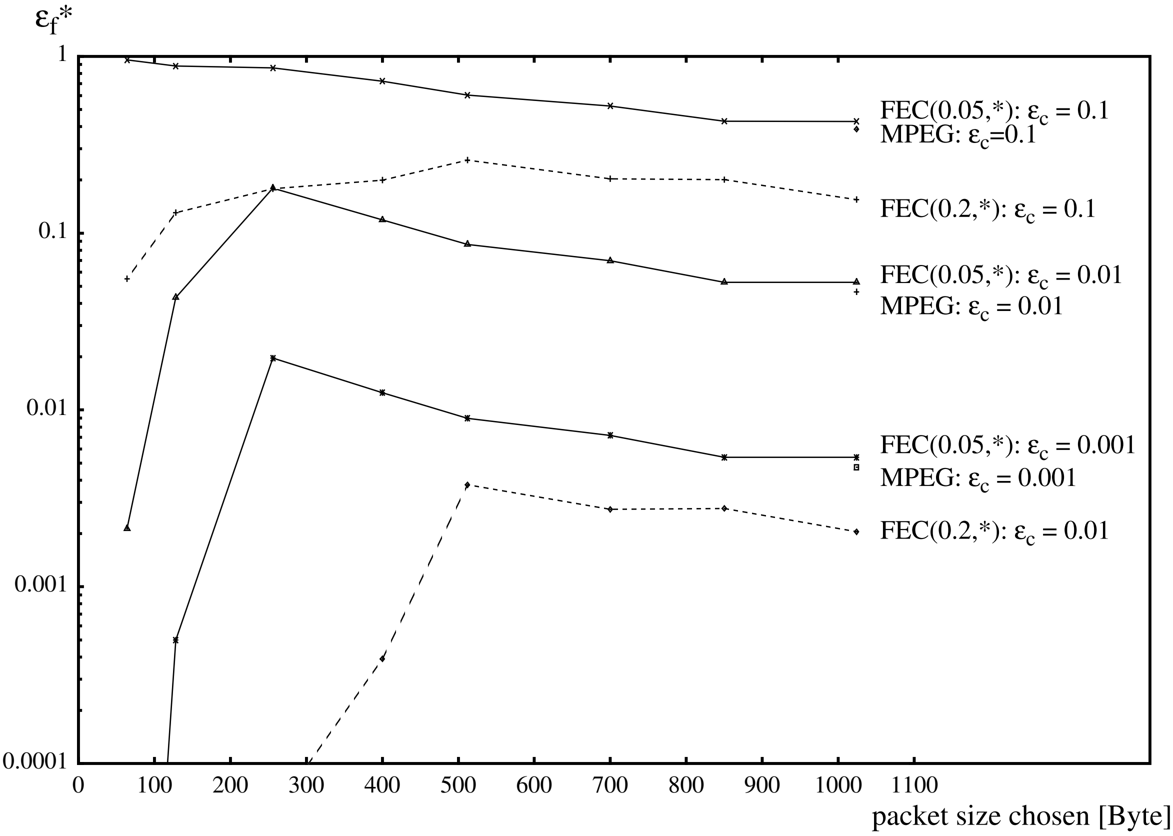

Fig. 8.1: Frame loss probability in dependence of the packet-size chosen for selected encoding variants

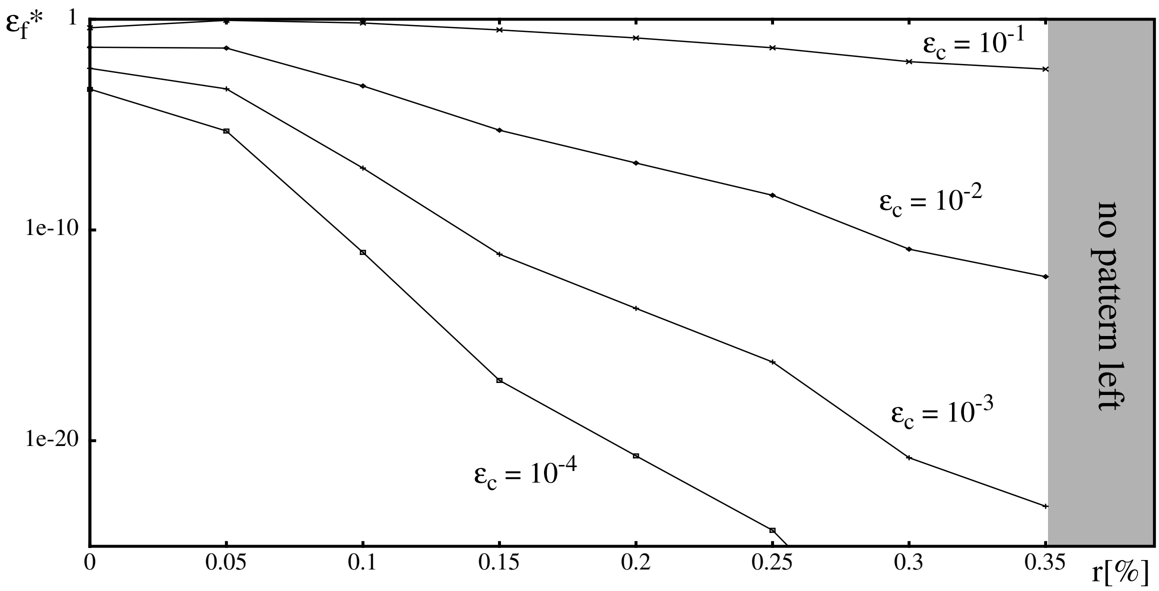

Fig 8.2: Frame loss probability in dependence of percentage of redundancy used for FEC