Session XI: Open Session

Chair: Radu-Pospescu Zeletin, Technical University of Berlin and GMD-FOKUS

The open session of the workshop had no talks scheduled. It served as the first part of a workshop conclusion, trying to identify the main multimedia research issues for the future. The session was based on short ad-hoc presentations and an open discussion on the different issues raised in Heidelberg.

The first issue discussed were new error recovery techniques in end-to-end multimedia communication. The main characteristics of end-to-end data communication for text and graphics is that high reliability is required. A large variety of techniques have been developed for this purpose in the past. Most of these techniques are based on hand-shaking protocols.

For audio and video data, one often sees the statement that they do not require error correction methods. This is not really always true, e.g., compressed video streams are quite sensitive to errors. Therefore, the question was raised if there exist mechanisms which provide reliability, but are based on unidirectional communication, thereby avoiding the delay problems inherent to handshaking.

Ernst Biersack from Bellcore presented results from experiments based on forward error correction (FEC) algorithms in an ATM environment. He integrates his mechanisms in the transport protocol (but in his own words, he "collapses everything between the application and the ATM interface into the transport layer"). Using FEC on a cell basis, a very low end-to-end error probability was achieved in the experiments presented. The induced redundancy in the data to be transported, however, affects the number of connections which can be supported.

FEC is a promising research topic due to the better integration of different types of information (data, voice, and video) with different characteristics and reliability requirements in one end-to-end stream. Ernst indicated that his first experiments showed that FEC works best for mixed scenarios of bursty and continuous traffic. It is also a promising approach for reliable multicast. However, FEC does not come for free – there are cases where the amount of redundancy is too high and others where it is too low. With selective retransmission, bandwidth is used to only the amount actually needed.

The second main topics of the session were flow control and synchronization. The discussion focused on which flow control mechanisms are needed for multimedia and how they can be combined with synchronization functions.

It was agreed that flow control on a rate basis is the best solution. Domenico Ferrari summarized this with: "Window flow control is out, rate control is in." For audio and video retrieval, some workshop participants mentioned that much, however, can also be achieved by output buffering and backpressure mechanisms. The induced delay may prevent the use of these mechanisms for conversational services.

During the workshop, different speakers have used the term "synchronization" with different meanings. During the open session we tried to clarify and accommodate different views and definitions. Ralf Guido Herrtwich presented a taxonomy of the different meanings and definitions of "synchronization" used during the workshop. On one hand, the term is used in upper layers of multimedia communication for expressing the required relation between events in a multimedia context. This synchronization is also called "synthetic synchronization" and is part of the multimedia scripting or document editing process. On the other hand, the term is used in the lower layers to express either the inter-stream or intra-stream synchronization. Intra-stream synchronization is also often called "jitter control" or "pacing." Ralf said that while inter-stream and intra-stream synchronization are different problems they may have a common solution (and presented the analogy to operating systems where semaphores are one common solution to both mutual exclusion and condition synchronization).

Low-level synchronization (inter- and intra-stream) was the subject of the following discussion. It was noted that several sources of timing exist in the system and that mechanisms have to be provided for an application to specify in relation to which time source synchronization shall be performed. Time sources may both be actual clocks and regular data sources. Referring to Ralf's previous statement of a common solution for low-level synchronization, Domenico Ferrari mentioned that with jitter control for single streams inter-stream synchronization comes for free.

Francois Horn from CNET concentrated on high-level synchronization. His definition of synchronization is that "any identified relation between events in a multimedia system constitutes a synchronization point." He identified temporal synchronization as a subset of a more general synchronization problem. To express synchronization one needs a language to describe these relations. He advocated for ESTERELLE as a language to be used in this context because of the precise formal specification that comes with it.

The last item addressed during the open session was the hardware architecture of future multimedia workstations. It was the general opinion that the multimedia workstation will be based on an internal switch as opposite to bus systems today. This is motivated mainly by the required processing concurrency in the multimedia environment. Andy Hopper made the remark that regardless of how the architecture really looks like, the most important thing is to avoid to change gear when communicating locally and remotely.

The open session provided the best forum during the workshop to elaborate on a wide variety of topics of common interest. Due to the intensive discussions in the audience it is recommended that future workshops should allocate more time for such type of events.

A Performance Study of Forward Error Correction in ATM Networks

Ernst W. Biersack
Bellcore
445 South Street
Morristown, NJ 07962-1910
E-mail: erbi@bellcore.com

November 1991

Abstract

The asynchronous transfer mode technique (ATM) has been standardized as the transport system for future broadband communication networks. ATM allows the multiplexing and transmission of data from different services over the same network. If the cell loss rate in the network is higher than the loss rate requested by the service, the protocols in the endsystems must make up for the difference in loss rate. Traditionally, a retransmission-based scheme (i.e. ARQ) is used to recover from loss of data. In high bandwidth delay-product networks the latency introduced by retransmission-based error recovery schemes may become unacceptably high. *Forward error correction* (FEC) schemes do not have this drawback. When FEC is used, the transmitter sends with the original data some redundant data that can be used by the receiver to reconstruct lost pieces of the original data without requesting their retransmission. FEC trades off an increase in the bandwidth required for the capability to recover from partial loss.

We use simulation to study an FEC scheme that can recover a fixed number of lost cells within a block of consecutive cells. The results obtained for different traffic scenarios demonstrate that FEC can reduce the loss rate by multiple orders of magnitude. FEC is shown to be most effective when used by video sources in a heterogeneous traffic scenario consisting of video and burst sources.

1 Introduction

Asynchronous Transfer Mode (ATM) is the internationally agreed upon transfer mode for broadband networks [ATM 89]. ATM provides high-bandwidth low-latency multiplexing and switching [MINZ 89]. The basic unit of multiplexing and switching in ATM is called a *cell.* A cell has a fixed length of 53 bytes: 48 bytes of data (*payload*) and 5 bytes of control information such as *virtual path identifier* (VPI), *virtual channel identifier* (VCI), and a *cyclic redundancy checksum* (CRC) based header error control. An ATM-type network will experience three types of errors: bit errors due to noise, switching errors due to undetected corruption of the cell header, and cell losses due to congestion. Losses due to congestion are expected to be far more common than the other two types of errors.

One of the reasons for adopting ATM is for the integration of services. ATM will be used by different applications that require services with widely varying *quality of service* (QOS) requirements. One QOS requirement is reliability of the transmission. If the degree of reliability provided by the network is lower than the reliability requested by an application, the endsystems must make up for the difference.

The two basic mechanisms available to improve reliability are *automatic repeat request* (ARQ) and *forward error correction* (FEC). ARQ is a *closed-loop* technique based on retransmission of data that were not correctly received by the receiver. ARQ requires the transmitter and receiver to exchange state information about the status of individual messages. Each retransmission of a messages adds at least one round-trip time of latency. Therefore, ARQ may not be applicable for transmitting data from applications with low latency constraints. Constrained latency services are necessary for human interaction, process control, remote sensing, etc. As the name implies, data for applications using this service is worthless if it does not arrive within a certain time. Services such as face-to-face audio or video require that the end-to-end latency be less than 100 milliseconds if the system is to provide acceptable performance. A connection across the continental USA could easily have a retransmission time of more than 100 milliseconds. Another disadvantage of ARQ based schemes is the complexity required to keep track of a potentially very large number of outstanding messages.

FEC is an alternative to ARQ that avoids its shortcomings and is well suited for operation in high bandwidth-delay product networks. FEC involves the transmission of redundant information along with the original data so that if some of the original data is lost, it can be reconstructed using the redundant information. In data communications, the use of FEC is attractive for supporting services that cannot rely on retransmission, such as real-time services over a high latency network. The amount of redundant information is typically small, so that FEC remains efficient. If ARQ is not feasible because of the additional latency, and the network itself does not have any other means of providing different streams with different QOS, FEC makes the operation of the network more cost-effective by allowing it to operate at a higher utilization. Without FEC, the network must be operated at a utilization where the loss rate never exceeds the most stringent loss rate required by an application. In this case, all applications would receive this low loss rate, independent of their actual need. FEC can also be used for an endpoint-based support of QOS. If a networks does not support different degrees of reliability, FEC is a viable mechanism to achieve the requested reliability.

The rest of the paper is organized as follows. The next section describes in detail the operation of FEC. Section 3 describes our experiment and the assumptions made. Section 4 presents the simulation results illustrating the effectiveness of FEC.

2 Operation of Forward Error Correction

Coding theory distinguishes two types of data corruption: an *error* is defined as a bit with an unknown value in an unknown location; whereas an *erasure* is a bit with an unknown value in a known location. If the FEC decoder is able to take advantage of erasure information, replacing an error with an erasure approximately doubles the error correcting power of the code. In ATM, one byte of the five byte header is a CRC for header error control that can correct one-bit errors and detect two-bit errors. For bit error rates smaller than 10^{-9}, which are typical for fiber optics transmission systems, the cell loss rate due to random bit errors is negligible. Therefore, congestion losses are the dominant form of error on ATM networks and the network can be modeled as a well-behaved erasure channel. Congestion losses are typically not random losses but then occur in bursts and often effect consecutive cells of a single stream. For FEC to be effective, it is important to recover the loss of multiple cells in a group of consecutive cells. Not only do congestion errors appear as erasures but occur in multiples of the cell size and align on cell boundaries. In order to be able to determine which cells are missing, cells must contain a sequence number. The FEC system used in this paper takes k cells as input and produces $k + h$ cells as output. At the receiver, any k of which is enough to recover the original information as long as none of the received cells is corrupted by bit errors (erasures only). A group of k cells from which the h redundant cells are generated is referred to as a *block*. The redundant cells are referred to as *over-code*. Advances in implementation technology make FEC possible even at speeds of several hundred Mb/s. At Bellcore, a Reed-Solomon-based code has been developed that has the desired performance and can be implemented on a single chip operating at 400 Mb/s with $h = 16$ or 1 Gb/s with $h = 4$ (and any k) [MCAU 90]. The FEC encoder and decoder are almost identical, which simplifies implementation. Figure 1 illustrates the operation of the FEC system for $k = 3$ and $h = 2$ cells[1]. The FEC encoder at the transmitter produces 2 redundant cells (fec-1, fec-2) for every block of 3 cells. In the example, one data cell (data-2) and one redundant cell (fec-2) get lost. The FEC decoder at the receiver uses fec-1 and the two correctly received data cells to reconstruct the missing data cell (data-2). The sum of the traffic generated by all sources is referred to as the (normalized) *load* λ. λ assumes values between 0 and 1, where $\lambda = 1$ means that cells are generated at the same rate as the multiplexer can handle them. Since FEC increases the total amount of traffic in the network by the number of redundant cells generated by the FEC encoder, the cell loss rate will increase whenever FEC is used. For FEC to be effective, it must recover enough lost cells to reduce the cell loss rate **after decoding** to a level lower than the cell loss rate when no FEC is used. FEC is applied by individual sources. A source that applies FEC is referred to as an *FEC source*, a source that does not apply FEC as a *non-FEC source*. If λ is the aggregated load generated by the sources, the load in the network after FEC is applied is referred to as the *effective load* λ_{eff}. We have $\lambda_{eff} = \lambda \times (1 + \frac{h}{k})$ if all

[1] Typically, the ratio k/h is much larger than in this example.

sources apply FEC.

A cell is a very small unit of data. To avoid the complexity of dealing with individual cells and to amortize the higher-layer headers over a larger unit of data, functions such as error detection and error correction are done over blocks of cells. We assume a block consists of k cells and is the unit of error detection and FEC. A block that has a least one cell missing/lost is said to be *corrupted.* When FEC is applied and h redundant cells are generated, a block is considered *lost* if more than h of its $k + h$ cells are lost. Our main measure for the performance of FEC will be the reduction of the block loss rate. When no FEC is applied the block loss rate $P_B(\lambda)$ is equal to *Pr(* $i > 0$ *cells are lost in a block of k cells* $|$ *load* λ*)*. When redundant cells are generated, the block loss rate **before** cell loss recovery at the receiver will rise to $P_B(\lambda_{\text{eff}})$. The block loss rate is a more meaningful measure of the network performance than the cell loss rate. It is, for instance, useful in estimating the expected losses in an ATM system that performs segmentation and reassembly of cells into blocks. The block loss rate **after** cell loss recovery was performed is referred to as $P_{Brec}(\lambda_{\text{eff}})$ and is equal to *Pr(*$i > h$ *cells are lost in a group of* $k + h$ *cells* $|$ *load* λ_{eff} *)*. The gain G due to FEC is defined as $G \stackrel{\text{def}}{=} \log_{10}(\frac{P_B(\lambda)}{P_{Brec}(\lambda_{eff})})$. G measures the reduction of the block loss rate in terms of orders of magnitude. ($G = j,\ j \in 0, 1, 2, \ldots,$ means that FEC reduces the block loss rate by j orders of magnitude.) For FEC to be effective, we require $G > 0$. When FEC is used only by a subset of the sources, the non-FEC sources will observe an increase in their block loss rate and their penalty D is defined as $D \stackrel{\text{def}}{=} \log_{10}(\frac{P_B(\lambda_{eff})}{P_B(\lambda)})$.

The following example illustrates the tradeoff between the increased cell loss rate and the recovery of lost cells due to FEC. Let us assume that the increase in load due to FEC causes the block loss rate to increase by a factor of ten, i.e. $P_B(\lambda_{\text{eff}}) = 10 * P_B(\lambda)$. For FEC to be effective and achieve $G > 0$ it must reconstruct enough cells to recover more than 90% of all corrupted blocks. If 99% of the corrupted blocks can be recovered, we have $P_{Brec}(\lambda_{\text{eff}}) = \frac{1}{100} * P_B(\lambda)$ and therefore $G = \log_{10}(10) = 1$, i.e. a block loss rate reduction by one order of magnitude over $P_B(\lambda)$.

3 Experimental Setup

To study the effect of FEC, we have built a simulator that models a multiplexer with N input ports and one output port (see figure2). The multiplexer is output-buffered with a single shared buffer of finite capacity B at the output port. Each cell time the multiplexer checks all inputs for a newly arriving cell and puts these cells in the output buffer, if there is any space, and drops them otherwise. To make the service to the different input ports fair, the multiplexer starts each cell time with a different input port: if at time T there is at least one cell arriving and the first port to be checked is port i, then at time $T + 1$ the first input port to be checked will be port $(i + 1)$*mod* N. Each input port has one source connected to it. Every source generates on the average the same amount of traffic. An ATM network will carry traffic from different types of applications with different statistics. In our simulation model we distinguish between two different types of sources: (1) *burst sources* representing applications such as bulk data transfer or transactions and (2) *video sources* representing variable bit rate (VBR) video sources such as entertainment video or video conferencing. A burst source is characterized by the ***interarrival time between bursts, burst size,*** and ***cell separation*** of cells within the burst. The interarrival time is geometrically distributed, the number of cells per burst is a constant, and the spacing between cells is fixed. A spacing of x means that during the transmission of a burst, a cell is transmitted every x cell times. The video source we use has been derived from entertainment video and therefore reflects the complex nature of this type of application. M. W. Garrett at Bellcore has generated the video data set by encoding a two hour-long action movie using an intra-frame 8×8 Discrete Cosine Transform coding scheme with run-length and Huffman encoding [GARR 91]. The data set contains the number of bytes per frame produced by the encoder. The duration of one frame is $\frac{1}{24}$ of a second, the total number of frames is 171,000. The statistics of this data — in terms of bandwidth — are: Maximum: 15.06 Mbit/sec, Mean: 5.34 Mbit/sec, Minimum: 1.79 Mbit/sec, and Maximum/Mean (burstiness): 2.82. In the simulator, the data for each frame is broken up into cells of 48 bytes and the transmission of the cells is spaced out equally over the duration of the frame. The different video sources are *unsynchronized*, i.e. they start at different points in the movie. Unsynchronized video sources exactly represent a situation where different people watch the same movie at different times (video on demand) and simulate the multiplexing of different video sources. The cells of a video source are spaced equidistantly within one frame-time. When FEC is applied to a video source, the spacing between cells is adjusted such that the original cells together with the redundant cells are spaced equidistantly within one frame-time. Since the video sources have a fixed bandwidth, the capacity of the multiplexer is altered to yield different values of λ. For a particular scenario, all sources generate the same amount of traffic.

Table 1 lists all the parameters of the simulation.

Parameter	*Value*
Load λ	0.7 – 0.95
Total number of sources N	32
Number of video sources *v_srcs*	24, 32
Number of burst sources *b_srcs*	8
Number of FEC sources *f_srcs*	0, 4, 8, 12, 16, 24, 32
Burst interarrival time	geometric distribution
Burst size	50 (cells)
Block length k	50 (cells)
Cell separation for video sources	equidistant over frametime
Cell separation *sep* for burst sources	10
Percentage $\frac{h}{k}$ of redundant cells per block	0, 10, 20, 30
Size of switch output buffer B	100 (cells)

Table 1: Simulation Parameters

4 Simulation study of FEC

The effectiveness of FEC depends very much on the cell loss behavior of the network, because FEC can only recover a limited number of lost cells per block. For FEC to work well, it is necessary that most corrupted blocks have less cells missing than the FEC-decoder can recover.

Since the traffic mix in future ATM networks is unknown, we use various homogeneous and heterogeneous traffic patterns to investigate the effect of FEC on the block loss rate. For each scenario we present various measures that illustrate certain key parameters. Let CL be a discrete random variable that represents the percentage of cells lost in block and can assume any value from $0, 1, \ldots, 100$. $F_{CL}(x) \stackrel{\text{def}}{=} Pr(CL \leq x \mid CL > 0)$ is the cumulative distribution function of the percentage of cells lost per block, provided at least one cell is lost. Therefore, $F_{CL}(x) = p$ means, that with probability p the percentage of lost cells in a corrupted block is less or equal than x. The larger the value of $F_{CL}(x)$ for a given x, the more corrupted blocks can be recovered with an over-code of x-percent. Another measure used in this experiment is the block loss rate $P_B(\lambda)$.

4.1 All video sources

Our first traffic scenario, referred to as **V1**, has 32 video sources that are all unsynchronized, a buffer size of 100, and a block size of 50. For different loads the percentiles of CL and the block loss rate $P_B(\lambda)$ are given in table 2. For a plot of $F_{CL}(x)$ see figure 3.

Load λ	**Block Loss Rate** $P_B(\lambda)$	90.0 **Percentiles of** CL	99.0	99.9
0.900	2.8e-06	4	6	6
0.925	2.6e-03	6	8	14
0.950	2.4e-02	6	12	24

Table 2: Scenario V1, all video sources.

Up to a load of $\lambda = 0.85$ no loss is observed. Increasing the load from 0.90 to 0.95 results in a sharp increase of the block loss rate by 10^4. The percentiles of CL show that most corrupted blocks lose only a small percentage of cells and a few redundant cells are sufficient to recover most corrupted blocks. At $\lambda = 0.925$, 90% of all corrupted blocks lose less or equal than 6% of the cells in a block (90.0 percentile), 99% of all corrupted blocks lose less or equal than 8% of the cells in a block (99.0 percentile), and 99.9% of all corrupted blocks lose less or equal than 14% of the cells in a block (99.9 percentile). However, when FEC is applied, the additional load due to the redundant cells will significantly increase the block loss rate of the non-FEC sources, imposing a high penalty on them. This is confirmed by a simulation with $\lambda = 0.90$ where 4 of the 32 video sources apply FEC with a 20% over-code (ten redundant cells per block of fifty cells). While the block loss rate $P_{Brec}(\lambda_{\text{eff}})$ for the FEC sources is reduced to zero the block loss rate $P_B(\lambda_{\text{eff}})$ for the non-FEC sources increases by a factor of 10^3, i.e. $D = 3$. When FEC is applied by more than 4 sources, FEC is not effective at all since $P_{Brec}(\lambda_{\text{eff}})$ is higher than $P_B(\lambda)$, i.e. $G < 0$. In general, the usefulness of FEC in a scenario with all video sources is limited. FEC can not be applied whenever the over-code necessary to achieve the required gain G causes the penalty D for the non-FEC sources to be higher than acceptable. Instead, the load must be kept low enough, to meet the most stringent loss requirement of any source.

4.2 Mixed Traffic Scenario

Future ATM networks will most likely carry data from different applications. Therefore, a more realistic traffic scenario is heterogeneous with 24 video sources and 8 burst sources. The parameters for the burst sources are burst length 50 and cell separation 10. The video sources are unsynchronized. This scenario is referred to as **VB24-8**. For VB24-8, the first losses are observed at $\lambda = 0.7$ as compared to $\lambda = 0.9$ where the first losses occurred for V1. For $\lambda \leq 0.90$, the cell losses perceived by the video sources are due to the interferences with the burst sources.

Figure 4 plots $F_{CL}(x)$ and table 3 gives the block loss rates and the percentiles of CL for the video sources. Not shown are the values for the burst sources. The block loss rates for the burst sources are the same as for the video sources. Due to the higher burstiness of the burst sources their percentiles of CL are significantly higher than the ones for the video sources. For VB24-8, independent of the load, 90% of all corrupted video blocks lose less or equal to 10% of their cells, 99% of the blocks never lose more than 20% of their cells. Therefore, for the same over-code, the gain due to FEC will be much higher when FEC is applied to the video sources in VB24-8 as compared to when FEC is applied to the burst sources in B1.

Load λ	**Block Loss Rate** $P_B(\lambda)$	90.00	99.00	99.90	99.99
		Percentiles of CL			
0.70	5.1e-05	4	8	10	12
0.75	4.0e-04	4	8	14	18
0.80	2.7e-03	6	10	16	24
0.85	1.3e-02	6	14	20	30
0.90	5.0e-02	8	18	26	36
0.95	1.4e-01	10	20	32	44

Table 3: Scenario VB24-8, Percentiles of CL for the video sources.

We performed additional simulations with FEC applied by some or all of the video sources. Scenario **VB-FEC-1** is derived from VB24-8 and has $\lambda = 0.7$. The over-code of the FEC-sources is 10%, the number of video sources applying FEC is 4, 8, 12, 16, or 24. Figure 5 shows how FEC effects the block loss rates $P_B(\lambda_{\text{eff}})$ and $P_{Brec}(\lambda_{\text{eff}})$. The block loss rate is given for each of the 32 sources individually. The station numbers 0–23 are allocated to the video sources, numbers 24–31 to the burst sources. When j stations apply FEC, their station numbers range from 0 to $j-1$. For VB-FEC-1 an over-code of 10% reduces the block loss rate $P_{Brec}(\lambda_{\text{eff}})$ for the eight FEC sources to $5 * 10^{-7}$, yielding a gain $G \geq 2$. Figure

6 plots the gain and the penalty for VB-FEC-1. Both, the gain and the penalty are linear functions of the number of sources that apply FEC. As the number of FEC sources increases, the penalty D increases. However, the penalty never exceeds $D = log_{10}(\frac{2*10^{-4}}{5*10^{-5}}) = 0.875$, which is noticeably smaller than the gain.

Scenario **VB-FEC-2** is derived from VB24-8 and has $\lambda = 0.8$. Four video sources use FEC and the amount of over-code is 10, 20, or 30 percent. Table 4 shows the effectiveness of FEC. When applied by four video sources, FEC reduces their block loss rate by about one order of magnitude per 10% over-code and has little impact on the block loss rate of the non-FEC sources.

Over-code	**Gain for Video FEC**	**Penalty for Video non-FEC**	**Penalty for Burst**
10 %	1.9	0.12	0.11
20 %	2.9	0.26	0.24
30 %	3.8	0.39	0.47

Table 4: Gain and Penalty for scenario VB-FEC-2.

We have seen that among all the scenarios investigated, FEC is most effective for the heterogeneous traffic scenario. Even for high loads, FEC can reduce the block loss rate by multiple orders of magnitude. The gain achieved for the FEC sources by far exceeds the penalty for the non-FEC.

5 Related work

Loss recovery using FEC has been studied previously [OHTA 91, SHAC 90, ZHAN 91]. Our work differentiates itself in several ways.
(1) Our traffic model for the video sources is derived from an actual source while previous models used to evaluate the performance of FEC assume that the inter-arrival times are exponentially or hyper-exponentially distributed.
(2) The code used to generate the redundant cells is more powerful. The redundant cells are generated via a modified Reed-Solomon code that is able to recover any h cells lost out of $k + h$ cells.
(3) Our main measure of performance is the block loss rate, while other studies focus on the cell loss rate.

6 Conclusion

We have investigated the performance of an FEC scheme that can recover up to h lost cells out of $k + h$ cells. The model consisted of N sources generating traffic for a multiplexer with a finite output buffer. The arrival for the burst sources follows a geometric distribution, the one for the video sources is derived from a real movie after it was encoded. We performed simulations with different traffic mixes to obtain the gain and penalty due to FEC. FEC was most effective for a scenario consisting of mixed burst and video sources. When applied by the video sources, FEC can reduce the block loss rate by many orders of magnitude, while the non-FEC sources are only marginally effected. For a homogeneous scenario, FEC is less effective. In the case of all video sources FEC is only effective if applied by a few sources and the penalty for the non-FEC sources is severe.

In summary, FEC can be very effective in reducing the block loss rate. The gain depends on the traffic scenario and varies with the load, the amount of over-code, and the number of FEC sources. If possible, FEC should be used only by a subset of the sources. For a fixed over-code, as the burstiness or the load increase, the gain FEC decreases.

Acknowledgment

I would like to thank M. W. Garrett for providing me with the data for the video source and C. J. Cotton for writing a first version of the simulator. The comments of C. J. Cotton, M. W. Garrett and A. McAuley on earlier versions of this paper are very much appreciated.

References

[ATM 89] Comité Consultatif International de Télégraphique et Téléphonique, "Broadband Aspects of ISDN", *Recommendation I.121*, 1989.

[GARR 91] M. W. Garrett and M. Vetterli, "Congestion Control Strategies for Packet Video", *Proc. 4th Int. Workshop on Packet Video*, Kyoto, Japan, August 1991.

[MCAU 90] A. J. McAuley, "Reliable Broadband Communications Using a Burst Erasure Correcting Code", *Proc. ACM SIGCOMM 90*, pp. 287–306, Philadelphia, PA, September 1990.

[MINZ 89] S. E. Minzer, "Broadband ISDN and Asynchronous Transfer Mode (ATM)", *IEEE Communications*, 27(9):17–24, September 1989.

[OHTA 91] H. Ohta and T. Kitami, "A Technique to Detect and Compensate Consecutive Cell Loss in ATM Networks", *Proc. IEEE INFOCOM '91*, pp. 781–790, Bal Harbour, FL, April 1991.

[SHAC 90] N. Shacham and P. McKenny, "Packet Recovery in High-Speed Networks using Coding", *Proc. INFOCOM 90*, pp. 124–131, San Francisco, CA, June 1990.

[ZHAN 91] L. Zhang and K. W. Sarkies, "Modelling of a Virtual Path and its Applications for Forward Error Recovery Coding Schemes in ATM Networks", *Proc. SICON '91*, Singapore, September 1991.

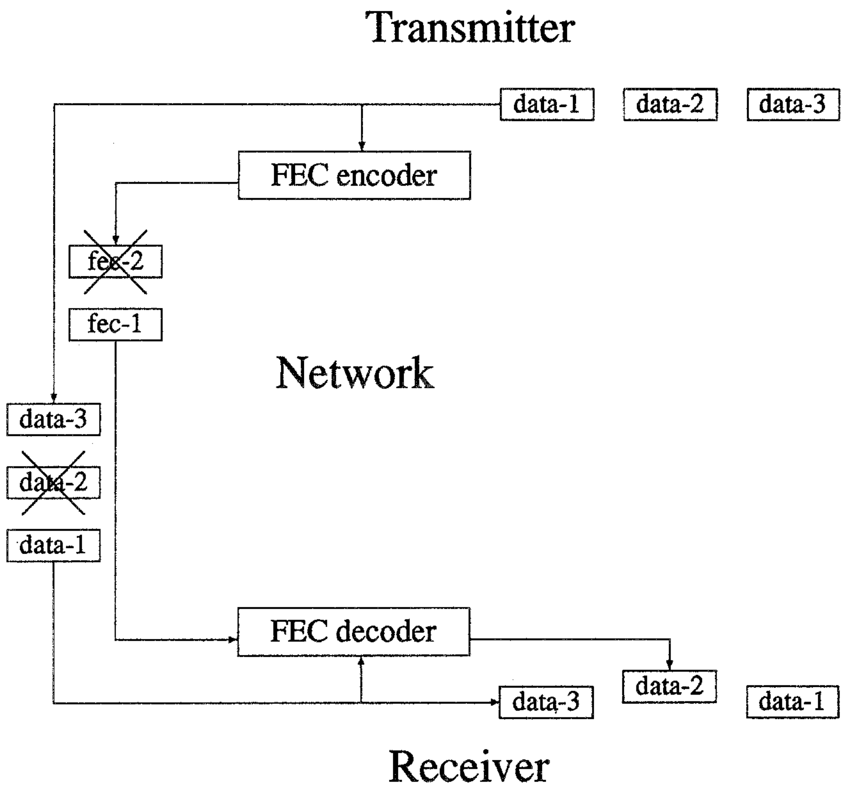

Figure 1: Operation of FEC.

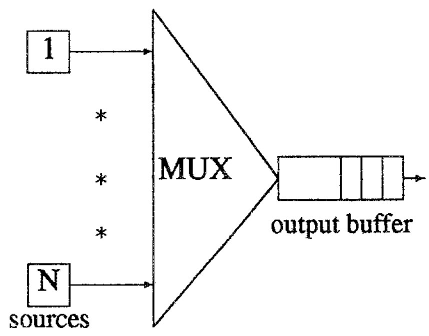

Figure 2: Model of the output-buffered multiplexer.

Cumulative Distribution Function of Percentage of Cells Lost per Block

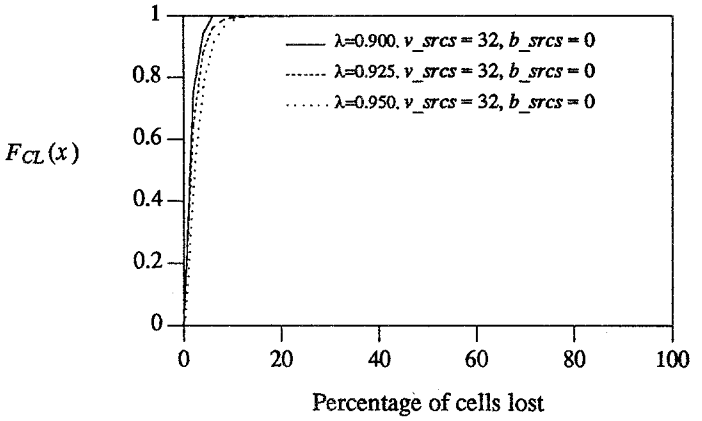

Figure 3: $F_{CL}(x)$ for scenario V1.

Cumulative Distribution Function of Percentage of Cells Lost per Block

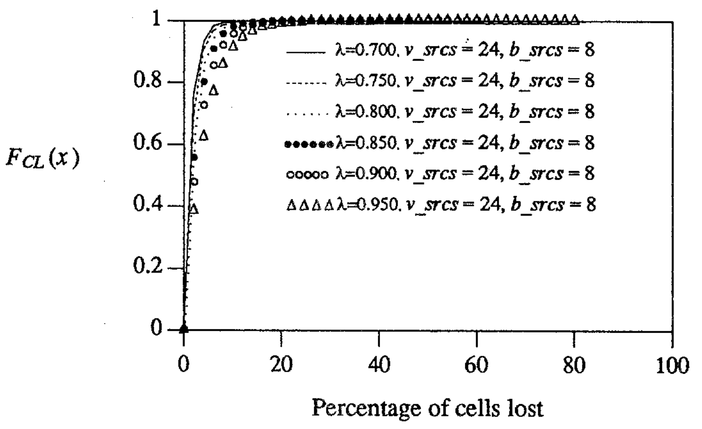

Figure 4: $F_{CL}(x)$ for the video sources in scenario VB24-8.

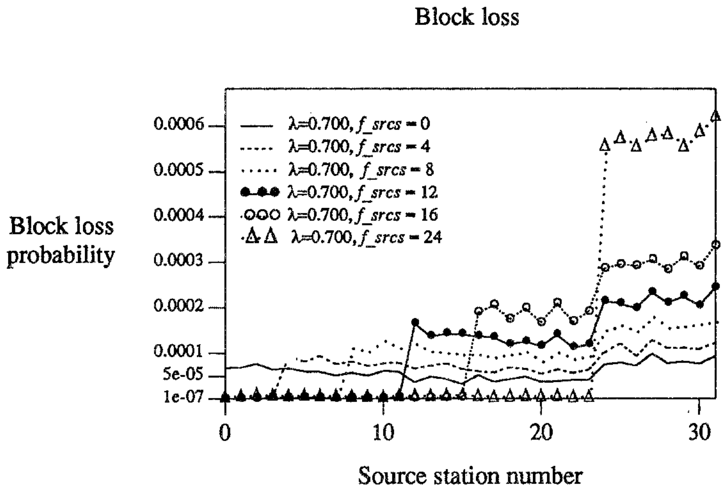

Figure 5: $F_{CL}(x)$ for scenario VB-FEC-1.

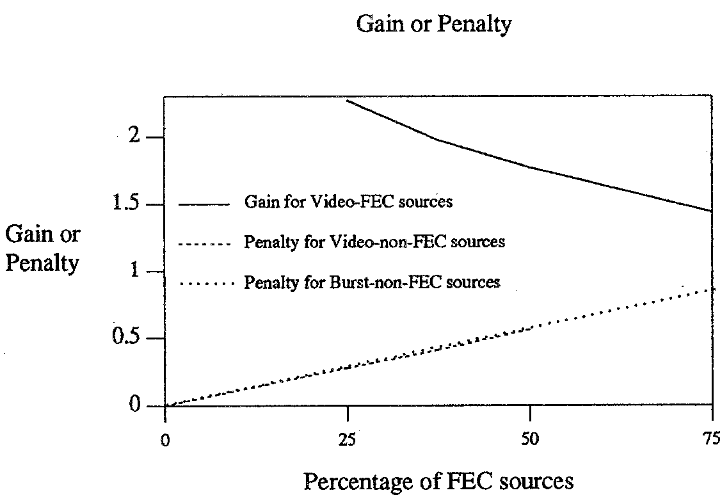

Figure 6: Gain and Penalty for scenario VB-FEC-1.