

# Multipath FEC Scheme for the ATM Adaptation Layer AAL5

*G. Carle*  
GMD FOKUS  
Kaiserin-Augusta-Allee 31  
Berlin, Germany  
carle@fokus.gmd.de

*S. Dresler, J. Schiller*  
Institute of Telematics  
University of Karlsruhe  
Karlsruhe, Germany  
j.schiller@ieee.org  
stefan.dresler@acm.org

## Abstract

One approach to facilitate statistical multiplexing of bursty sources in ATM networks is dispersion of the traffic over independent paths. It has been shown within the literature that by dispersion of traffic over disjoint paths it is possible to increase the Quality of Service (QoS) (see e.g., (Maxemchuck, 1993), (Lee, 1993)). In multipath schemes without additional error control, the resulting QoS depends on the worst-case path. By using a multipath scheme with FEC (Forward Error Correction), it is possible to compensate cell losses. Using error correcting codes like Reed-Solomon-Erasure Codes (RSE Codes, (McAuley, 1990)), these schemes add redundancy and disperse the traffic using several transmission paths (Ding, 1995).

Until now, protocols and implementation concepts suitable for a widespread use of multipath schemes are still missing, in particular for the widely used adaptation layer AAL5. We present a novel protocol MP-FEC-SSCS for AAL5 for employing a redundant multipath communication scheme especially suited to the UBR (Unspecified Bit-Rate) service. This presentation is complemented by detailed results for software implementations based on C++ and first results using Java, targeted for an active network implementation.

## Keywords

ATM, AAL, SCS, FEC, Reed-Solomon-Erasure Code, Striping, Multipath Communication Scheme, Dispersion Routing

## 1 INTRODUCTION

ATM networks are designed to allow an efficient use of network resources by statistical multiplexing of different types of traffic streams, such as data, audio, and video communication. A key challenge arises from the need to handle both highly bursty and unpredictable sources, along with relatively smooth sources. In the presence of highly bursty sources it proved to be difficult to achieve a statistical multiplexing gain while ensuring a high quality of service (QoS) for all connections.

One approach to facilitate statistical multiplexing of bursty sources is to use a multipath communication scheme and to disperse the traffic over independent paths. It has been shown within the literature that by dispersion of traffic over disjoint paths it is possible to increase the Quality of Service (QoS) (see e.g., (Maxemchuck, 1993), (Lee, 1993)). This improvement of QoS is possible because splitting of traffic leads to a reduced burstiness of the individual substreams. By using a Forward Error Correction (FEC) scheme that transmits redundant data over one or more additional paths, a receiver is able to reconstruct the original data even in case of losses. This is particularly attractive as ATM services with lower reliability can be provided at lower costs. Walker et. al. (Walker, 1997) e.g. propose a set of ATM tariffs, where the price per volume for VBR (Variable Bit Rate) traffic is two orders of magnitude higher than the price per volume for UBR (Unspecified Bit Rate) traffic. This price difference gives a high incentive to select the UBR service class whenever possible. While UBR services frequently will show high cell loss rates, the combination of UBR services with an FEC scheme allows to achieve relatively high reliability with very low costs.

### 1.1 AAL-level realization of a Multipath FEC Scheme

Dispersion of traffic over multiple paths and adding of redundancy can be realized within an ATM network at different layers. The FEC scheme of (Ohta, 1991) for example has been designed for being applied at VP (Virtual Path) level. An integration of a multipath FEC scheme at the ATM layer would require hardware modifications of switches and end systems. An alternative approach of integrating a multipath FEC scheme above the Common Part Convergence Sublayer (CPCS) of AAL5 can be realized in software, without hardware modifications.

Reducing the effective bit error rate and cell loss ratio for the layer above the adaptation layer (e.g., IP) will significantly improve the overall quality of service. This is especially true for applications that require the provision of a highly reliable service. A number of previous researchers (Shacham, 1990), (Ohta, 1991), (Biersack, 1993), (Ayanoglu, 1993) have shown how the use of FEC can improve end-to-end ATM performance in terms of effective throughput and latency.

### 1.2 Potential gain by striping

The potential gain obtained by using a striping mechanism is explained by the following example. Figure 1 shows an application sending a message with a size of six cells. From these six cells, three redundancy cells are derived using an error correcting code. This is achieved by lining up cells in a matrix consisting of  $m$  ( $m=2$  here) rows of length  $n$  ( $n=3$  here). Choosing the number of redundancy rows to be  $h=k-m=3-2=1$ , a single row of

redundancy cells is computed. In our example, a simple code like an XOR code is sufficient which computes the bitwise XOR of each column in the matrix. User as well as redundancy cells are then transmitted on, say, three paths, e.g., by assigning each path a row of the matrix. Different assignments are possible, each with its characteristic strengths suited to specific loss assumptions. The receiver reconstructs lost user cells and reassembles the original message. Details on this process and the cell and frame format used are explained below.

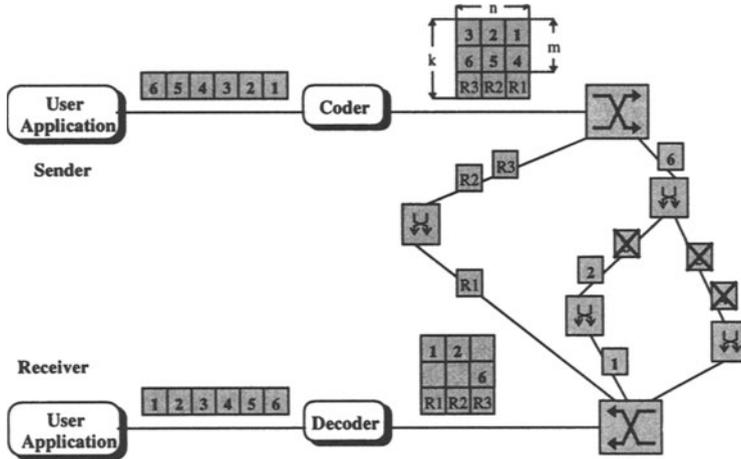


Figure 1 Multipath communication in ATM Networks.

By appropriately choosing the parameters it is possible to construct a coding scheme that allows for the reconstruction of user messages even if one of the paths suffers from the loss of all cells belonging to the message, as long as no other cells are lost.

## 2 PROTOCOL ISSUES

This chapter presents details on the protocol on which our redundant multipath communication scheme is based.

### 2.1 Multipath-FEC-SSCS

Data transmissions over ATM currently usually use AAL5 due to its small bandwidth overhead for protocol fields, and due to its low processing requirements. It was chosen to develop the multipath FEC scheme as a Service Specific Convergence Sublayer (SSCS) of the ATM Adaptation Layer AAL5 and to name it MP-FEC-SSCS.

For reconstruction of lost cells belonging to an AAL frame it is necessary to identify cells by cell sequence numbers. Assuming that no successful reconstruction of a frame will be required in cases where more than 15 cells get lost consecutively on a single path, a cell sequence numbers of 4 bits (counting modulo 16) can be used. If all parameters (number of columns, user data rows and redundancy rows, etc.) of an SSCS session are fixed, no additional per-frame information fields are necessary. Adding an

FEC-SSCS frame header containing the parameters of a session provides for additional flexibility, however, at the cost of a small bandwidth and protocol processing overhead.

Figure 2 gives an overview of the PDU structure used in the Multipath-FEC-SSCS protocol. User data forms the upper part of a matrix, and redundancy is located in the lower part of the matrix. Every matrix element has a length of 46 bytes, leaving 2 bytes of the payload of an ATM cell for per-cell protocol information of MP-FEC-SSCS. This protocol information, called the MP-FEC-SSCS cell header, contains the 4 bit cell sequence number, one bit which is alternated between successive frames to allow for a simple receiver state machine, one bit indicating whether a cell contains user or redundancy data, and a 10 bit checksum (CRC-10).

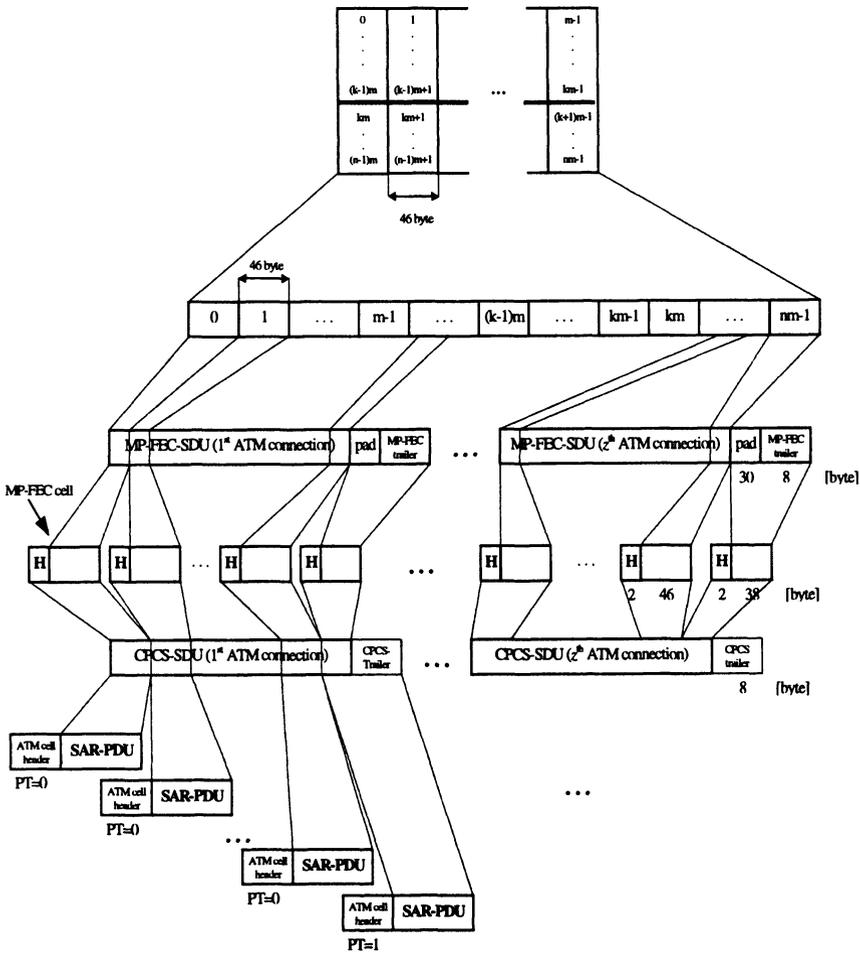


Figure 2 Mapping of MP-FEC-PDUs onto ATM Cells.

The last cell of an MP-FEC frame consist of the 2 byte MP-FEC-SSCS cell header, a 30 byte SSSCS padding field, an 8 byte MP-FEC-SSCS-Trailer, and the 8 byte AAL5-CPCS-Trailer. The MP-FEC-Trailer contains fields for the number of MP-FEC cells in a matrix, the number of redundancy cells per matrix, a frame sequence number, and the length of the AAL-SDU in bytes. Also included is a bit indicating the type of data carried (user/redundancy data or peer-to-peer signaling information).

### 3 PERFORMANCE EVALUATION

This chapter presents a performance evaluation of MP-FEC-SSCS by a theoretical analysis, based on simulation, and based on implementations in C++ and Java.

#### 3.1 Theoretical gain achievable by multipath schemes

Existing results presented by (Maxemchuck, 1993), (Lee, 1993), (Ding, 1995), (Brendan, 1995), (Adishesu, 1996) already proved that multipath communication schemes with and without FEC can reduce latency and cell loss probability for bursty sources. For a multipath scheme without FEC, the dispersion of bursty traffic over multiple paths allows to reduce cell loss, as the probability for queue overflow at an individual switch is reduced in comparison with a non-parallel scheme. For a multipath scheme with FEC, the additionally transmitted redundancy cells increase the load and are therefore responsible for an increased cell loss probability in comparison with a multipath scheme without FEC. However, in many cases the reduction of cell loss because of dispersion of the traffic over multiple paths will be significantly larger than the increase of cell loss due to additional redundancy cells of a multipath FEC scheme.

Figure 3 shows an example for the cell loss probability of a non-parallel scheme and a multipath communication scheme with and without FEC, as obtained by simulations in (Ding, 1995). The figure shows the observed cell loss probability for a scenario where the traffic is splitted at a first switch, transmitted over several switches, aggregated at a last switch, and finally decoded, corrected, and resequenced at the receiver. Curves are given for no redundancy cells ( $R=1.0$ ), 10% redundancy ( $R=1.1$ ), and 20% ( $R=1.2$ ).

The loss probability is evaluated for on-off sources with exponentially distributed on- and off-states, an average batch size of 200 cells, a peak rate of 150 Mbit/s, and for switch output buffers with 800 cells. In the case of the multipath schemes, the data cells of every burst are transmitted over 5 different paths.

The following formulas show how adding redundancy significantly increases the probability that an AAL5 frame is successfully delivered at the receiver. The cell loss probability of the individual links is called  $q_i$ . Without FEC, the probability for successful delivery of a frame with a length of  $k$  cells is given by

$$1 - Q = \prod_{i=1}^N (1 - q_i)^k, \quad (1)$$

with  $Q$  denoting the frame loss probability. For identical cell loss probabilities  $q = q_i$  on all links, (1) can be simplified to

$$1 - Q = (1 - q)^k. \quad (2)$$

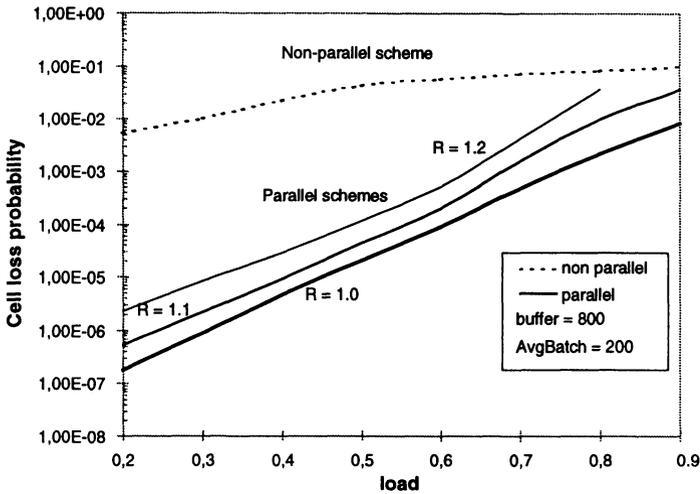


Figure 3 Cell loss probability as function of the load.

This calculation assumes statistically independent cell losses. Independent losses lead to a higher frame loss probability  $Q$  than statistically dependent cell losses. The assumption of cell losses occurring in a single burst per frame or at least in different columns of the coding matrix — based on the observation that cell losses occur in bursts (Ohta, 1991) — leads to the following estimation of the frame loss probability. In case of a multipath communication scheme with FEC,  $k$  information cells are protected by  $h$  redundancy cells. The frame loss probability decreases from the frame loss probability  $Q(k)$  of a scheme without FEC to  $Q(n,k)$  for the hybrid scheme, with the length of a frame with  $h$  redundancy cells being  $n = k + h$  cells. The frame loss probability  $Q(n,k)$  can be evaluated to

$$Q(n,k) = 1 - \sum_{i=0}^h \binom{n}{i} q^i (1-q)^{n-i} = \sum_{i=h+1}^n \binom{n}{i} q^i (1-q)^{n-i} . \tag{3}$$

Figure 4 shows the impact of FEC on the resulting frame error rate for AAL5 frames with a payload of 7.5 Kbyte, equivalent to 160 data cells ( $k=160$ ), and for a redundancy of 0 ( $R=1.0$ ), 10% ( $R=1.1$ ), 20% ( $R=1.2$ ) and 30% ( $R=1.3$ ). For the example shown in Figure 3, the additional redundancy cells of a multipath FEC scheme with 20% redundancy increases the cell loss rate by approximately an order of magnitude. At the same time, the FEC scheme reduces the resulting frame loss rate already for 10% redundancy and up to a cell loss rate of  $10^{-3}$  by more than two orders of magnitude, even more for 20% redundancy.

Deployment of FEC in multipath scenarios has thus shown to be useful in order to reduce the frame loss rate at the receiver(s). The slightly larger amount of traffic for FEC is dispersed over several connections.

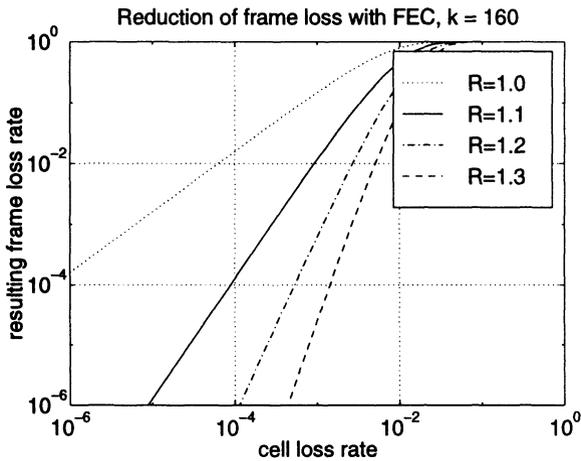


Figure 4 Impact of FEC onto resulting frame error rate.

### 3.2 Implementation Issues

MP-FEC-SSCS has been implemented using object-oriented programming techniques on the basis of C++ and the runtime library Channels 1.0 developed by Siemens (Böcking, 1995), using a SUN SPARC 10 with SunOS 4.1.3. The ATM connections were emulated by socket communication on top of IP over ATM. A SUN SPARC 20 workstation running Solaris 2.5.1 linked back-to-back with the SunOS station was used for emulation of an ATM switch dropping cells with predefined loss characteristics.

In addition to the implementation using the programming language C++, we implemented the performance critical FEC coding and decoding functions using Java. While today the performance of Java is not satisfying, the promising properties like platform independence and built-in security can be useful already. One example is the use of MP-FEC-SSCS within the Active Network approach, where nodes inside the network perform computation besides the pure routing and forwarding, such as coding and billing (Active Networks, 1997). Using Java, our code can also be pushed onto such nodes making the dissemination of the protocol much simpler. The performance problems of today's Java implementations can be overcome using specialized Java processors or the ever increasing power of standard processors together with just-in-time compilers (IEEE, 1997).

### 3.3 MP-FEC-SSCS Processing Delay

The processing delay introduced by the respective functions are listed in Table 1. The numbers are the maximum and minimum values of 100 measurements. The long time to receive an MP-FEC cell and to store it in the buffer is due to non-optimal memory allocation at the receiver. The table shows that the sender side needs 94 to 110  $\mu$ s to create an MP-FEC message, and the receiver side 163 to 468  $\mu$ s to process an incoming MP-FEC message.

Table 1 Execution times for sending and receiving

		<i>execution time (<math>\mu</math>s)</i>	
		<i>min</i>	<i>max</i>
sender	generate MP-FEC cell header	23	25
	generate MP-FEC cell (incl. cell header)	40	50
	generate MP-FEC trailer	31	35
receiver	detection of MP-FEC-PDU boundaries (per cell)	20	25
	reassemble MP-FEC cell and store in receive buffer	71	355
	decode and reassemble cell header incl. detection and localization of cell loss	20	35
	delete redundancy from cell matrix and store AAL-SDU in buffer	52	53

For two redundancy rows per matrix, amounting to 10% redundancy, and 6 connections, the coding and decoding delays are shown in Figure 5.

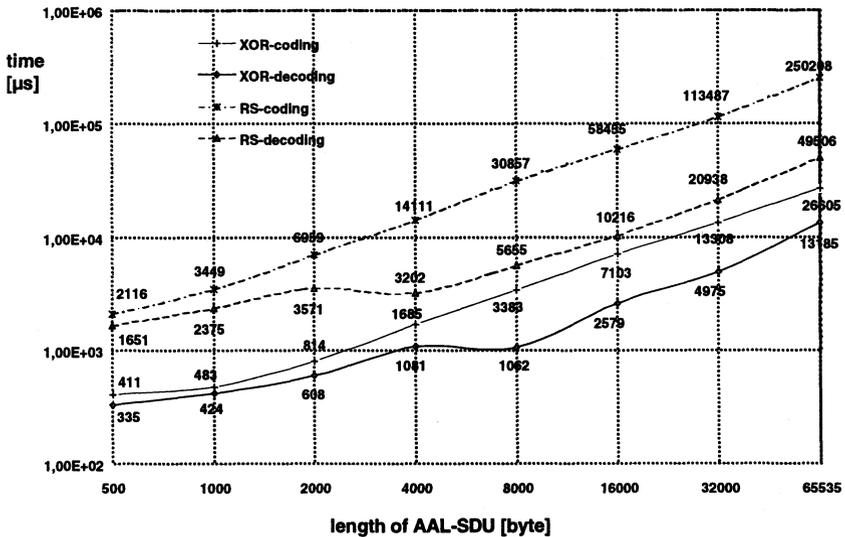


Figure 5 Time for en-/decoding using XOR and Reed-Solomon coding (C++).

The time to encode or decode a message increases (in large parts linearly) with its size in bytes, for both the simpler (and faster) XOR coding scheme, and the more powerful (and slower) RSE code. Encoding takes more time on average since it is always

necessary to compute the redundancy cells at the sender, whereas the decoding process only has to be invoked if cells of a message are missing, and then only on those columns of the matrix that are affected. If a sender and a receiver are located on the same host and an SDU size of 65535 byte is chosen, usage of the RSE coding slows down transmission to at most  $1 / ((250208 \mu\text{s} + 49508 \mu\text{s})/65535 \text{ byte}) = 218.7 \text{ kbytes/sec}$ .

The XOR coding achieves a rate of up to  $1 / ((26605 \mu\text{s} + 13185 \mu\text{s})/65535 \text{ byte}) = 1.647 \text{ Mbytes/sec}$  under these conditions (SUN Sparc10, SunOS 4.1.3).

Implementing the RSE coding in Java on a PentiumPro with 200 MHz, transmission rates of up to 260 kbit/s can be achieved with the just-in-time compiler. Using the interpreted version of Java, only the much simpler XOR-scheme reaches 280 kbit/s loading the CPU completely. Main reasons today are a very inefficient memory management and limited capabilities for fast copying of data.

Figure 6 shows the time needed for encoding and decoding using Reed-Solomon and XOR for 8000 byte AAL-SDUs depending on the redundancy vector  $h$ . In comparison with XOR it can be shown that Reed-Solomon coding is not the better solution for software implementations in praxis. With a growing  $h$  the average encoding and decoding time also increases drastically, while the times for XOR increase only slightly.

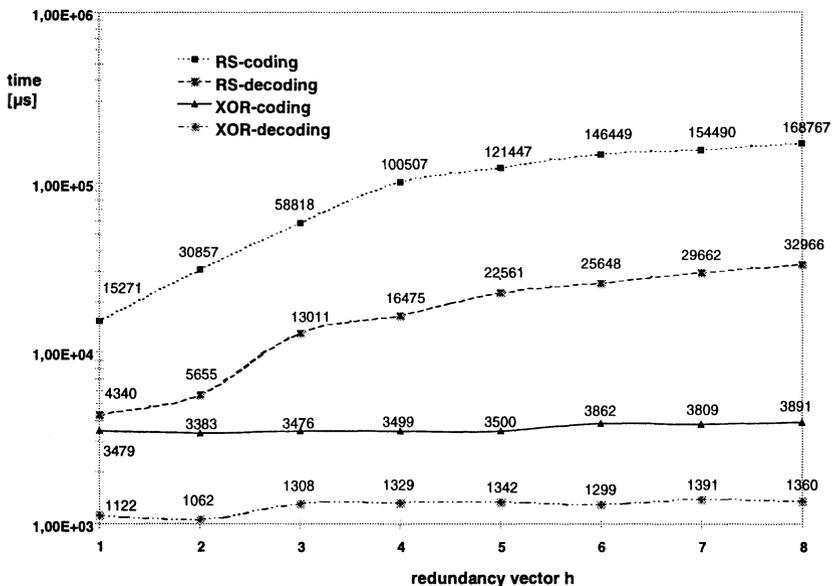


Figure 6 Time for coding and decoding depending on the redundancy vector.

Using a two-state Markov approach to model the loss characteristics of an ATM connection, a simulation was performed to obtain the probability of a successful reconstruction of a coding matrix at the receiver side. The probabilities  $P_1 =$  "a successfully received cell is followed by another successfully received cell" and  $P_2 =$  "a

lost cell is followed by another lost cell" were varied from connection 1 (with  $(P_1, P_2) = (0.97, 0.7)$ ), corresponding to a cell loss rate of  $9.1e-2 = 9.1\%$  to connection 7 (with  $(P_1, P_2) = (0.9997, 0.3)$ ), corresponding to a cell loss rate of  $4.23e-4$ .

Again, it can be seen that RSE codes are more powerful than XOR schemes. Furthermore, it can be derived from the figure that in scenarios of connections with differing loss the multipath FEC scheme is still able to recover a substantial amount of messages that otherwise would have been erroneous because at least one cell of it got lost.

A protocol that introduces too much overhead compensates the gain it achieves by increasing the network load beyond a useful point. As described above, MP-FEC-SSCS adds an MP-FEC trailer and a 40 byte padding to each MP-FEC-SDU, i.e., for each connection. Furthermore, it adds cell headers of 2 bytes each to every 46 bytes of a message. Finally, AAL5 adds its CPCS-Trailer to each CPCS-SDU. Figure 7 shows the protocol overhead depending on the number of parallel connections and on the length of the AAL-SDU for a redundancy amount  $r = 0.1$ , i.e., the value of the quotient of redundancy cells and total cells. For long SDUs, the overhead approaches  $r$ . An AAL-SDU of size 16000 byte across 4 parallel connections has an overhead of approx. 15.6% with redundancy chosen as  $r=0.1$ .

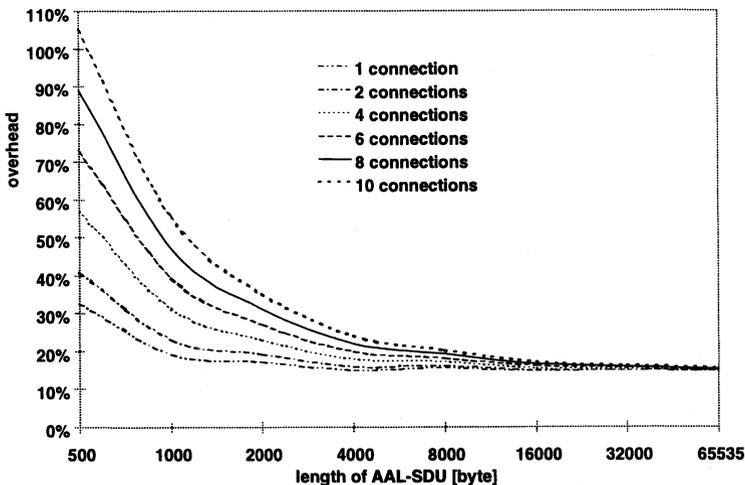


Figure 7 Protocol overhead depending on the number of parallel connections.

### 3.4 Resulting tradeoffs

Using redundancy, whether on a single connection or on several, involves increased bandwidth requirements compared to unsecured transmission. Figure 3 showed that a multipath transmission protocol experiences an increase in cell loss by about a magnitude if 20% of redundancy is added. This increase is more than compensated for by the cell reconstruction abilities of MP-FEC-SSCS, as shown in Figure 4. An AAL-SDU of 8000 byte length, corresponding to 167 cells in AAL5, e.g., has an overhead of

less than 20% (compare Figure 8) in MP-FEC-SSCS while at the same time the protocol is able to reconstruct most of the cells lost (Figure 6).

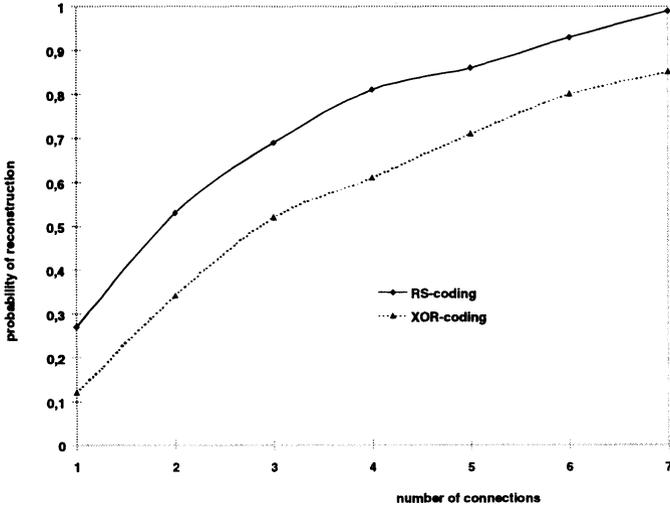


Figure 8 Probability of reconstruction depending on the number of connections.

#### 4 CONCLUSIONS

Applications with bursty traffic characteristics that require certain QoS like a maximum delay and an upper bound for the error rate may significantly benefit from using a multipath FEC scheme. In contrast to error control with retransmissions, FEC allows for error control with low delay. By dispersion of traffic over several connections, both the burstiness of a traffic source can be reduced, and the negative impact of additional traffic of an FEC scheme can be compensated, and the benefits of FEC for error recovery can be fully exploited. The MP-FEC-SSCS described has shown to increase the QoS while keeping the network overhead small. Together with the platform independence of the Java implementation, dissemination of the approach within Active Networks is doable in the near future. With the implementation platforms used, only the C++ implementation allows to achieve the desired performance and proofs to be readily applicable.

#### 5 REFERENCES

- Active Networks, <http://www.sds.lcs.mit.edu/darpa-activenet/>
- Adishesu, H., Parulkar, G., Varghese, G. (1996) A Reliable and Scalable Striping Protocol. Proceedings of ACM SIGCOMM Conference, Vol. 26, No. 4
- Ayanoglu, E., Gitlin, R.D., Oguz, N.C. (1993) Performance Improvement in Broadband Networks Using Forward Error Correction, *Journal of High Speed Networks*, 2, 287-304.

- Biersack, E. (1993) Performance Evaluation of Forward Error Correction in an ATM Environment. *IEEE Journal on Selected Areas in Communication*, **11(4)**, 631-640
- Böcking, S., Seidel, V., Vindeby, P. (1995) CHANNELS - A Run-Time System for Multimedia Protocols. 4th Intl. Conference on Computer Communications and Networks (ICCCN), Las Vegas.
- Brendan, C., Traw, S., Smith, J. (1995) Striping Within the Network Subsystem. *IEEE Network*, **4**
- Carle, G., Dresler, S. (1996) High Performance Group Communication Services in ATM Networks, in *High-Speed Networks for Multimedia Applications* (eds. W. Effelsberg, O. Spaniol, A. Danthine, D. Ferrari), Kluwer Academic Publishers, Boston/Dordrecht/London
- Carle, G., Schiller, J. (1995) Enabling High Bandwidth Applications by High-Performance Multicast Transfer Protocol Processing. 6th IFIP Conference on Performance of Computer Networks, PCN95, Istanbul
- Carle, G., Esaki, H., Guha, A., Dwight, T. (1995) Necessity of Cell-Level FEC Scheme for ATM Networks. ATM Forum Technical Committee ATMF95-0325
- Carle, G., Esaki, H., Guha, A., Dwight, T., Tsunoda, K., Kanai, K. (1995) Draft Proposal for Specification of FEC-SSCS for AAL Type 5. ATM Forum Technical Committee ATMF95-0326
- Ding, Q., Liew, S. (1995) A Performance Analysis of a Parallel Communications scheme for ATM Networks. *IEEE Globecom*
- Gustavson E., Karlsson G. (1997) A Literature Survey on Traffic Dispersion. *IEEE Network*, **2**, 28-36
- IEEE Micro (1997) Making Java work. *IEEE Micro features Java*, **3**
- Lee, T., Liew, S. (1993) Parallel Communication for ATM Network Control and Management. *IEEE Globecom*
- Lin, S., Costello, D. (1983) Error Control Coding: Fundamentals and Applications. Prentice Hall, New Jersey
- Maffeis, S. (1997) iBUS – The JAVA Intranet Software Bus. Technical Report, Olsen& Associates Zurich
- Maxemchuk, N. (1993) Dispersity Routing on ATM Networks. *IEEE Infocom*, San Francisco, pp. 347-357
- McAuley, A.J. (1990) Reliable Broadband Communication Using a Burst Erasure Correcting Code. *ACM SIGCOMM*, pp. 297-306
- Ohta, H., Kitami, T. (1991) A Cell Loss Recovery Method Using FEC in ATM Networks. *IEEE JSAC*, **9**, 1471-1483
- Shacham, N., McKenny, P. (1990) Packet Recovery in high-speed networks using FEC coding. *IEEE INFOCOM*, San Francisco
- Walker, D., Kelly, F., Solomon, J. (1997) Tariffing in the New IP/ATM Environment. *Telecommunications Policy*, **21**, 283-295