# An Object Oriented Approach for Evaluating the Error Correction Coding

R Shiva Shankar[1], P. Neelima,[2] V. Priyadarshini [3], D Ravi Babu[4]
[1-4] Department of CSE, S.R.K.R Engg. College, Affiliated to Andhra University, Bhimavaram, W.G.District, Pin-534 204, A.P. INDIA,

***Abstract*:-A Model-based analytic mechanism was proposed for estimating the overall efficiency of forward error correction coding (FEC) joined with interleaving in combating packet losses in the IP networks. By fully making use of forward error correction coding the loss of several packets throughout the data transmission can be decreased. By forming the network path in relation with the single multiplexer model, an approach was developed for the precise assessment of the packet-loss information for processes arrived in general and also developed an easy procedure for the most complex multiple-session scenario. This combined approach offers an integrated framework for exploring the tradeoffs among the important coding parameters such as interleaving depths, channel coding rates and block lengths. This system also delivers the information on the performance possible with forward error correction coding in the IP networks.**

***Keywords: forward error correction coding, interleaving, packet-loss***



## I. INTRODUCTION

The packet transportation service offered by representative packet-switched networks, as well as IP networks, is not consistent and the quality-of-service (QoS) cannot be assured. Because of buffer overflow in switching nodes packets may possibly lost, be rejected by reason of extreme bit errors and failure to pass the cyclic redundancy check (CRC) at link layer, or also discarded by network control techniques as a reply to congestion occurred someplace in the network. Forward error correction (FEC) coding has frequently being proposed for endwise recovery from packet losses mention above. But, the use of FEC in this application delivers a double-edged sword. From the view of an end user, FEC can help recover the packets that are lost in a timely fashion over the use of redundant packets, and usually adding extra redundancy can likely to increase performance provided, this extra added redundancy cannot affect the network packet loss features badly. Alternatively, from the viewpoint of network, the extensive use of FEC schemes by the end nodes will raise the raw packet-loss rate in the network as a consequence of the extra loads resulting from broadcast of redundant packets. Thus, in order to improve the endwise performance, the suitable tradeoff, in terms of the quantity of added redundancy, and its effect on network packet-loss procedures, requires to be examined under particular and realistic modeling expectations.

The whole efficiency of packet-level FEC coding, employing interlocked Reed-Solomon codes, in combating network packet losses and deliver an information theoretic methodology in order to determine the optimum cooperation between end-to-end performance and the related increase in raw packet-loss rates by using a accurate model-based analytic method. For a given chance of block length we expect that there is a finest choice of redundancy, or channel coding rate, as a rate also too high (low redundancy) is only not capable enough to efficiently recover packet losses while a rate is too low (high redundancy) results in extreme raw packet losses due to the increase in overhead which overrate the packet recovery abilities of the FEC code. The best channel coding rate results in an optimum compromise among the set two effects.

The use of redundant parity packets was recommended to reform lost data packets and the consistent performance estimation denotes that residual packet-loss rates can be decreased up to three orders of degree. But, in [4] for analysis purposes the packet-loss procedure resulting from the single-multiplexer model was supposed to be free and, so, the simulation results offered visualize that this simplified study overestimates the performance of FEC.

## II. LITERATURE SURVEY

Cidon et al. [5] proposed a recursive algorithm to measure the packet-loss information (block error density), over which the precise residual packet-loss rate next to decoding was calculated. Unexpectedly, all numerical results presented in [5] denoted that the resultant residual packet-loss rates with coding are at all times more than without coding, therefore FEC is unsuccessful in this application. But, in [4] simply a single parity packet is used and the block length is restricted to the range. These decisions are slightly misleading and result from unsuitable parameter selections. In [6]-[8], more common arrival processes were taken into account and coding tradeoffs evaluated but performance results were found using large deviation bounds to characterize the packet-loss procedures.

In conflicting to [4], [5], cidon and rest used the frame-loss probability as the estimation metric for performance FEC, since it was supposed that the failure to recover any data packet lost will tend to the loss of every data packet in that definite block. For ATM networks, this presumption is effective because the loss of a one cell does results in the removal of the entire message. However, this is not the situation with other networks, such as IP networks. In IP networks, packet-level FEC coding can be implemented through various IP packets. Even Though data packets that are lost cannot be recovered, the properly received data packets in the identical coding block may still be helpful. For instance, in the application of video above IP, when some of the lost video packets cannot be recovered, the properly received video packets in the identical coding block required not be rejected and can be used to evaluate the information in lost video packets using an suitable error concealment structure.

## III. RELATED WORK

### A. *Forward Error Correction*

FEC is established by the addition of redundancy to the information to be transferred using a prearranged algorithm. Every redundant bit is invariably a difficult function of several actual information bits [8]. The actual information may or may not present in the encoded output; codes that contain the unchanged input in the output are **systematic**, whereas those that do not are **nonsystematic**.

A very easy example would be an analog to digital converter that trials three bits of signal asset data for every bit of transferred data. The easiest illustration of error correction is for the receiver to suppose the correct output is provided by the most regularly happening value in every group of three.

Forward Error Correction (FEC) is a kind of error correction which improves on simple error detection schemes by allowing the receiver to right errors once they are identified. This reduces the need for retransmissions.

FEC does well by the addition check bits to the leaving data stream. Addition of extra check bits decreases the amount of obtainable bandwidth, but also permits the receiver to correct many errors.

Forward Error Correction is specifically well matched for satellite transmissions, where bandwidth is realistic but latency is important.

### B. *Interleaving:*

In computer science interleaving is used to organize data in a non-contiguous way so as to increase performance. It was used in:

- Time-division multiplexing (TDM) in telecommunications.
- Computer memory
- disk storage

Interleaving is mostly used in data communication, multimedia file setups, radio transmission (for example in satellites) or by ADSL. The word multiplexing is occasionally used to talk about the interleaving of digital signal data.

### C. ***Interleaving in disk storage***

The major determination of interleaving is to correct the timing changes between when the computer was set to transfer data, and when that data was really received at the drive head in order to read. Interleaving was mainly used to organize the sectors in more effective manner probable, thus next to reading a sector, time would be allowed for processing, and then the next sector in arrangement is arranged to be read just as a computer is set to do so. Matching the sector interleave to the processing speed therefore rushes the data transfer, but an improper interleave can create the system perform noticeably slower.

### D. ***Interleaving in data transmission***

Interleaving in digital data transmission technology is used to guard the broadcast against burst errors. These errors modify more number of bits in a row, so a usual error correction pattern that assumes errors to be more evenly dispersed can be overwhelmed. Interleaving is used to support stop this being happened.

Regularly data is broadcast with error control bits that allow the receiver to correct a definite number of errors that exist during transmission. If burst error exists, too many errors can be prepared as one code word, and that codeword cannot be appropriately decoded. To decrease the effect this kind of burst errors, the bits of a many codeword's are interleaved prior to broadcast. In this manner, a burst error affects merely a correctable quantity of bits in every codeword, and then decoder can decode the codeword's properly.

This approach is widespread due to its less difficulty and inexpensive way to handle burst errors than straightforwardly increasing the strength of the error correction pattern.

.

### E. ***Disadvantages of interleaving***

Usage of interleaving methods in turn increases the latency. This is due to the whole interleaved block should be received prior to the crucial data is returned.

## IV. MODULES:

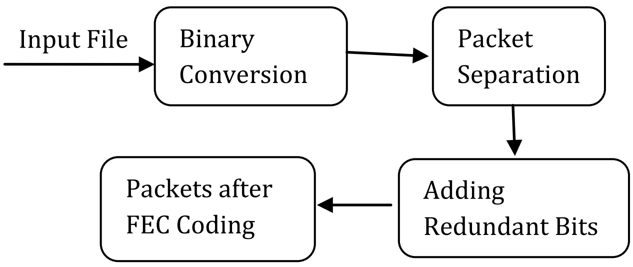
The modules are

- FEC Encoder
- Interleaver
- Implementation of the Queue

- De-Interleaver
- FEC Decoder
- Performance Evaluation

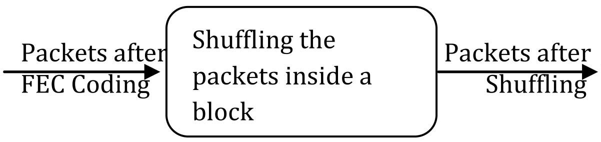
→FEC Encoder

In this module we add some redundant data to the specified input data, called FEC Encoding. The text exist in the input text file is transformed into binary. The binary conversion is done for all characters in that input file. Then we assign the redundant data for every bit of the binary. After the addition of redundancy we have a block of packets for every character.



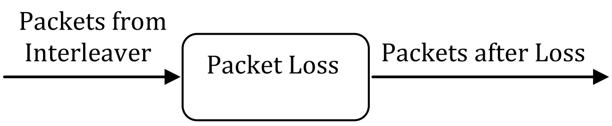

→ Interleaver

This module accepts the input as chunk of bits from the FEC Encoder. In this module we rearrange the bits within a single block in order to alter burst errors as random errors. This packet rearrangement procedure is completed for all blocks coming from the FEC Encoder.



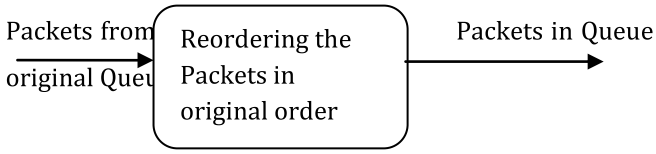

→ Implementation of the Queue

This module accepts the data packets from the Source system. This data is in blocks after FEC Encoding and Interleaving procedures are completed. Once we obtain the packets from Source, we form a packet loss. It is the method of removing the packets randomly. Next to the creation of packet loss we send the left over blocks to the Destination.



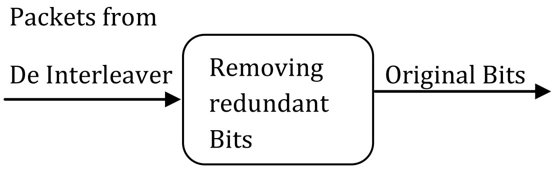

→ De-Interleaver

This module gets the blocks of data from a Queue. In this module we organize the data packets within the block in the same order prior to Interleaving. The procedure of Interleaving and De-Interleaving is completed to transform burst errors as random errors. Next to De-Interleaving the blocks are set in the actual order. Then the chunk of data is given to the FEC Decoder.



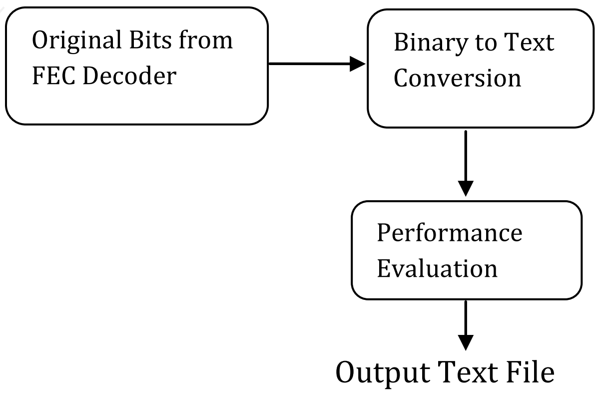

→ FEC Decoder:

This module accepts the input from the De-Interleaver. The obtained packets are progressed to eliminate the actual bits from it. So, we recover the actual bits of a character present in this module. Later retrieving the actual bits, we change it to characters and write it within the text file.



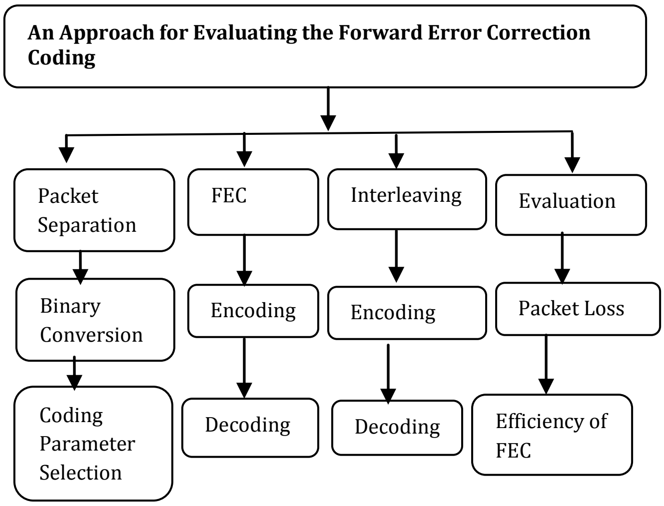

→ Performance Evolution:

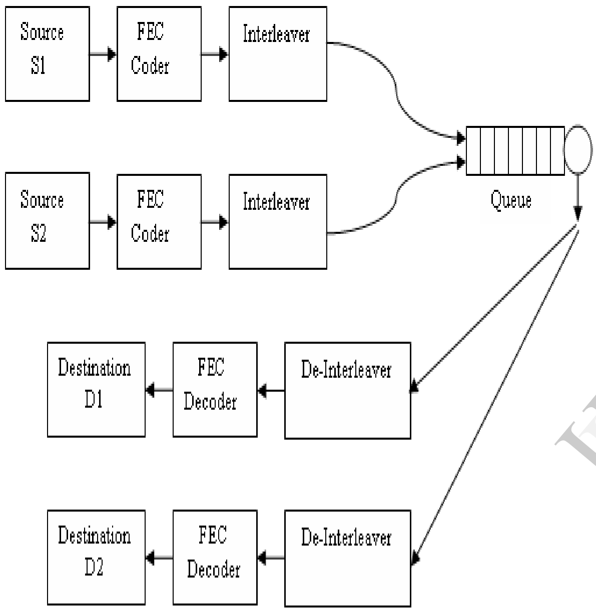
In this module we compute the total performance of FEC Coding in recovering the lost packets. Later retrieving the actual bits, we change it to characters and write it within the text file. This performance is computed by using the coding parameters such as Coding rate, Interleaving depth, Block length and various other parameters. First we compute the quantity of packet loss and then we use many formulas to compute the total performance of Forward Error Correction in the recovery of network packet losses.




## V. SYSTEM ARCHITECTURE

System Architecture illustrates "the entire structure of the system and the paths in which that structure delivers conceptual integrity". Here, the file be uploaded by source, then the uploaded file was encoded with FEC Encoding. In FEC encoding byte format is altered to binary and the addition of redundant data it is completed. Here the data is separated into minor sized packets. Then it is processed to interleaving in which these packets are interchanged in non-contiguous manner. Currently, these packets are processed to destination user via Queue. If there is any packet loss they are refilled by using the redundant data packets. The end user at destination will de-interleave and de-encoded the packets and get the actual message without having any loss in it.




*A. ALGORITHM:*

**Start FEC Encoding with the following steps:**

While (i< FILELENGTH)
Print a FileByte in Binary format;
Separation of each binary value into 2-D String array
Adding redundant Data to each Binary values
Merging final FEC binary values for interleaving process

**Start interleaving process by the following Steps:**

Do
For each value 'j' in merged File array
Copy the character value into TransferFile array at index 'i';
i ← i+1;
oD(while(i< FILELENGTH))

Prepare Final TransferFile array for each packet using shuffleChar array
Do
For each value 'j' in shuffleChar array
Add the character value into transferFile array at index 'i';
i ← i+1;
oD(while(i< FILELENGTH))

**Sending packets to Destination:**

Create InputStream for QueueAddress.txt file
While(!EOF)
Add each character value to destData and trim data value if necessary;
While end;
Print the Address of Destination;
Create Socket connection to Destination'1';
Send the Packet Value to the Destination '1';
Send the Packets to the Destination'1'.

This is the algorithm for process packets from the source to destination. Next to uploading a file FEC Encoding begins in the system then interleaving will in turn starts, send the packets to specific destination by make use of a Queue.

The Queue contains all the files that are required to be routed from source to destination. After completion of queue sending files to the exact destination then the vice-versa procedure will happen in the system there by in turn generate the actual text file.

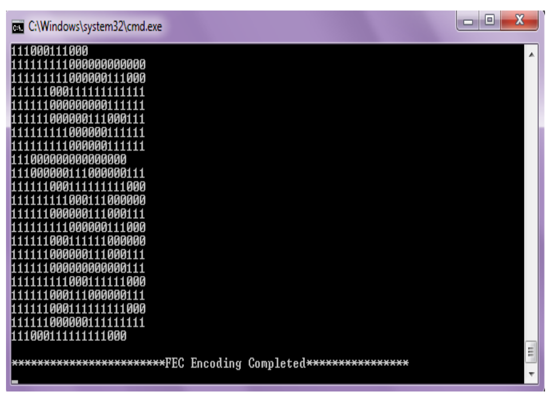


## VI. RESULT ANALYSIS:

### ADVANTAGES

The benefit of forward error correction is that a back-channel is not needed, or that retransmission of data can be frequently avoided, at the cost of greater bandwidth necessities on an average. FEC is thus applied in the cases where retransmissions are fairly costly or unbearable. In specific, FEC information is generally added to many mass storage devices to guard against destruction to the data stored.

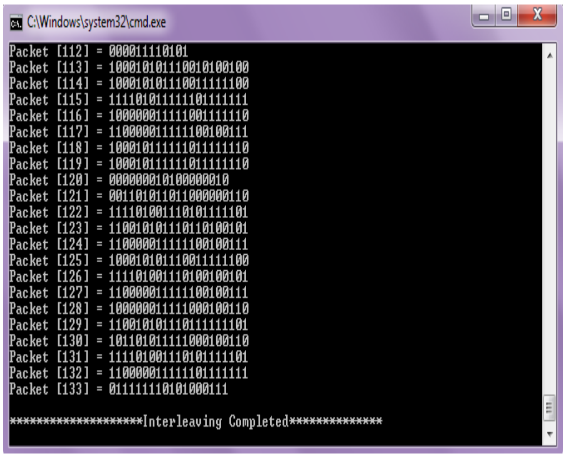
### FEC ENCODING:



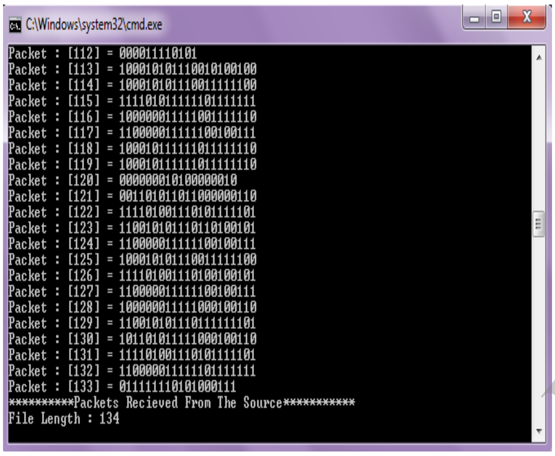

Binary format for a file that is uploaded into the system with repeated 1's and 0's.

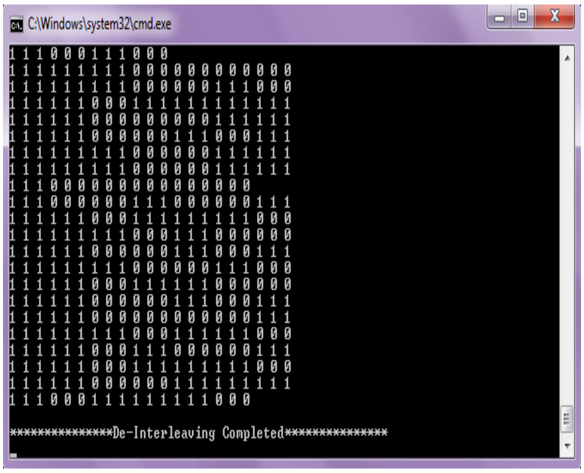
ISSN: 2278-0181
Vol. 3 Issue 4, April - 2014

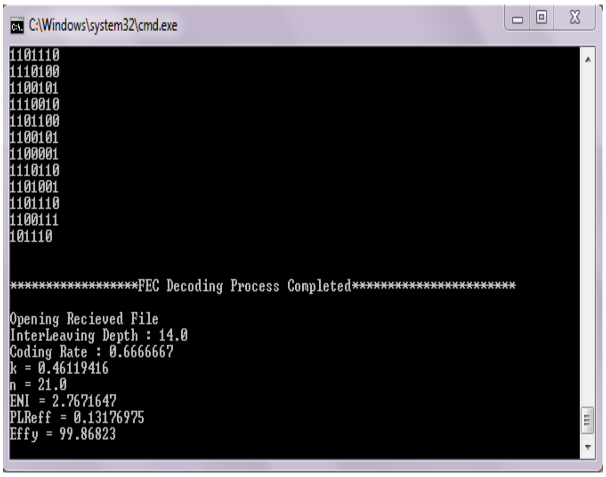
INTERLEAVING:




The Packets has shuffled bits with repeated bits in it. This is called INTERLEAVING.



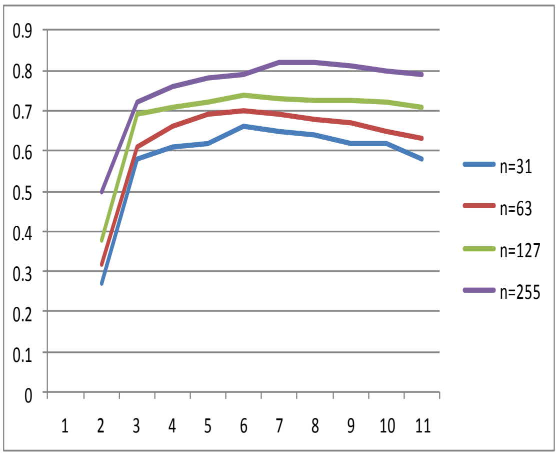




APPLICATIONS

FEC is used in ATM networks for the data broadcasting. So by make use of FEC for data broadcast in ATM networks we can get back the packets that are lost. Forward Error Correction (FEC) permits recovery from packet loss. The experimental results visualizes that the proper channel coding rate is needed go acquire the optimal results of Forward Error Correction coding (FEC)



The below graph illustrates the performance of FEC with several sources are multiplexed. Here the load from the every source is limited with the buffer size as 10 means that the buffer can contain up to 10 packets.




Channel Coding Rate ($R_c$)

The channel coding Rate $R_c$=k/n, and it is experienced that total count of sources multiplexed grows when there is an rise in channel coding block size(n).With FEC coding great number of sources multiplexed grows when a proper channel coding rate ($R_c$) is selected. To attain greater multiplexing coding larger block size is needed.

## VII. CONCLUSION

The efficiency of FEC in combating network packet losses depending on the single-multiplexer network model was examined and proved that FEC has higher potential in getting the packets that are lost initiated by congestion in a packet-switched network, providing that the coding rate and subsequent coding parameters are properly selected to achieve the optimal performance of FEC. We also verified a model for how much interleaving depth is needed for forward error correction coding (FEC) to attain the optimum performance.

## REFERENCES:

[1]. O. Ait-Hellal, E. Altman, and A. Jean- arie, Calculating loss probabilities in an M/M/1 queue with redundant retransmissions," presented at 33rd Annual Allerton Conference On Communication, Control, and Computing, Illinois, September 1995.
[2]. C. Partridge, *Gigabit Networking.* Reading, MA: Addison-Wesley, 1993.
[3]. "A measurement-analytic approach for QoS estimation in a network based on the dominant time scale," *IEEE/ACM Trans. Networking*, vol. 11, no. 2, pp. 222–235, Apr. 2003.
[4]. D. Y. Eun and N. B. Shroff, "Network decomposition: Theory and practice," *IEEE /ACM Trans. Networking*, vol. 13, no. 3, pp. 526–539, Jun. 2005.
[5]. J. Bolot, "End-to-end delay and loss behavior in the Internet," in *Proc. ACM SIGCOMM 1993*, San Francisco, CA, Sep. 1993, pp. 289–298.
[6]. N. Shacham and P. Mckenney, "Packet recovery in high-speed networks using coding and buffer management," in *Proc. IEEE INFOCOM 1990*, San Francisco, CA, Jun. 1990, vol. 1, pp. 124–131.
[7]. R. Kurceren, "Joint source-channel coding approach to transport of digital video on lossy networks," Ph.D. dissertation, Rensselaer Polytechnic Inst., Troy, NY, May 2001.
[8]. R. Kurceren and J. W. Modestino, "Optimum FEC coding rate allocation for video transport over ATM networks," in *Proc. IEEE Int. Symp. Information Theory (ISIT 1998)*, Cambridge, MA, Aug. 1998, p. 251.
[9]. R. Kurceren and J. W. Modestino, "A joint source-channel coding approach to scalable delivery of digital video over ATM networks," in *Proc. Int. Conf. Image Processing (ICIP 2000)*, Vancouver, Canada, Sep. 2000, vol. 1, pp. 371–374.
[10]. A. Jean-Marie, P. Dube, D. Artiges, and E. Altman, "Decreasing loss probabilities by redundancy and interleaving: A queueing analysis," in *Proc. ITC 18*, Berlin, Germany, Sep. 2003.