A Simulation Study of Two-Level Forward Error Correction for Lost Packet Recovery in B-ISDN/ATM

Nihat Cem Oğuz
Electrical and Electronics Eng. Dept.,
Bilkent University, Ankara, 06533, Turkey

Ender Ayanoğlu
AT&T Bell Laboratories
Holmdel, NJ, 07733-3030, USA

Abstract

The major source of errors in B-ISDN/ATM systems is expected to be buffer overflow during congested conditions, resulting in lost packets. A single lost or errored ATM cell will cause retransmission of the entire packet data unit (PDU) that it belongs to. The performance of the end-to-end system can be made much less sensitive to cell loss by means of forward error correction. In this paper, we present the results of a simulation study for an ATM network where forward error correction is performed at both the cell level and the PDU level. The results indicate that *(i)* cell losses are highly correlated in time, and analytical models ignoring this fact will not yield accurate results, *(ii)* the correlation of cell losses is similar to burst errors in digital communication, and similar code interleaving techniques should be used, *(iii)* coding cells and PDUs separately provides this interleaving effect, and this joint code outperforms coding only at the cell level or only at the PDU level in almost all cases simulated.

1 Introduction

In high-speed integrated packet-switched networks such as the *Broadband Integrated Services Digital Network* (B-ISDN) with the *Asynchronous Transfer Mode* (ATM) packet protocol, the end-to-end propagation delay for a typical connection will be much larger than the duration of a packet. Consequently, retransmissions associated with the conventional error detection and *Automatic Repeat reQuest* (ARQ) mechanisms will cause degradation in the delay-throughput performance. In ARQ, each retransmission increments the delay of a packet by approximately the round-trip propagation time. This is intolerable for many high-speed applications, especially for those sensitive to both loss and delay, such as distributed processing and interactive computing. The problem of a large propagation delay with respect to the packet size is present in satellite and deep-space communications, where error correction techniques are employed to increase reliability. In a similar manner, it has been suggested to use Forward Error Correction (FEC) to improve reliability without increasing the end-to-end delay in high-speed networks [1]–[7].

The basic idea in FEC is to add redundant information to the original data so that the receiver can recover lost information using this redundancy, and hence, avoid retransmissions. In competition with this recovery capability, however, there is an opposite effect of FEC at work: adding redundancy to the original data increases the load in the network, and in turn, the loss rate. FEC can be useful only when the former effect prevails.

In this work, we simulated a long-distance connection through an ATM network, and quantified the improvement in delay-throughput performance achieved by using FEC. In ATM, the basic unit of transport, switching, and queueing is a 53-byte *cell*: 48 bytes of payload and 5 bytes of header.Cells are grouped into variable size packets, also known as *Packet Data Units* (PDUs), at the ATM adaptation layer. While passing through the network, some of the cells are lost at congested switches. Therefore, some of the PDUs arrive at the destination with missing cells. By transmitting parity cells along with the information-bearing ones, cell losses in some PDUs can be recovered. A PDU is considered lost if its missing cells cannot be recovered. Normally, the transmitter is informed of the lost PDUs, and they are retransmitted. Our principal motivation is the fact that the burstiness in cell losses strongly affects the performance of FEC. Therefore, in addition to coding over consecutive cells, which is effective when cell losses are dispersed evenly or are "random," we propose coding over PDUs. Our results indicate this is quite effective in the case of burst cell losses.

2 Proposed FEC Technique

In coding over consecutive cells, the encoder appends M_A independent parity cells for each group of N_A consecutive information-bearing cells. The receiver determines the position of losses in this block of $N_A + M_A$ cells by means of sequence numbers. Hence, the end-to-end connection can be viewed as an ***erasure channel***. It is then possible to design an optimal, maximum distance code so that the decoder can recover the whole block provided that it arrives with less than or equal to M_A *erased* cells [8]. For example, in [9], Ayanoğlu *et al.* considered an optimal, maximum distance separable code based on either a Fourier-Galois transform or a Reed-Solomon code for self-healing communication networks. We consider this block of $N_A + M_A$ cells as a PDU. We call a PDU *coded* if $M_A > 0$ and *uncoded* otherwise.

Due to the statistical multiplexing in ATM networks, queues at the switching nodes may occasionally be congested, during which time cells are subject to a high probability of loss. In such cases, it is possible that more than

0-7803-0950-2/93/$3.00©1993IEEE

M_A erasures hit a coded PDU. To resolve this difficulty, in addition to coding over consecutive cells, we employ a similar code over PDUs: each block of N_P information-bearing PDUs, coded or not, is followed by M_P independent parity PDUs. Then, similarly to the case of cell coding, it suffices for the decoder to receive any N_P PDUs out of $N_P + M_P$ to recover the whole coding block. We call this block of $N_P + M_P$ PDUs a *coding block*.

Observe that there are $k = N_P M_A + M_P N_A + M_P M_A$ parity cells used for $N_P N_A$ information-bearing cells. The optimal code with the same parameters is the one that can recover any pattern of up to k erased cells out of $(N_P + M_P)(N_A + M_A)$. However, the decoding delays may then be too large. With the proposed technique, the recovery capability is structurally distributed over subblocks (PDUs) in the whole coding block, and hence, we take the advantage of faster decoding at the expense of losing decoding flexibility.

3 Simulation Model

We consider a long-distance *Virtual Channel* (VC) connection through an ATM network. The VC consists of 4 intermediate nodes and 5 links of length equivalent to 2048 *slots*, where a slot is the unit time needed to serve a cell at any standard ATM transmission speed. In each one of the intermediate nodes, there is a non-blocking 8 × 8 ATM switch capable of transporting all the simultaneous input cells to the requested output ports in zero time. We consider two *output queueing* techniques. In the first technique, there is a reserved buffer of capacity B cells for each output port (*complete partitioning*). In the second technique, all the cells to be queued share a common buffer pool of capacity $8B$ cells regardless of the output requests (*complete sharing*). We consider cell losses due to buffer overflows only. Although complete partitioning avoids interaction of traffic streams destined for different output ports, and yields smaller queueing delays as compared to complete sharing, the latter technique is expected to provide savings in cell losses when the traffic is bursty [10].

We concentrate on the forward traffic flowing from the source to the destination through the VC, and perform FEC on this *tagged* traffic. At intermediate nodes, the tagged traffic interferes with the *untagged* cells belonging to other VCs. We assume that PDUs of N_A cells arrive at the tagged and untagged sources continuously according to independent Poisson processes with rate p/N_A, where p is the normalized load offered by a source. The cells of a PDU are transmitted consecutively. The untagged PDUs join the tagged VC with probability 1/8, and the untagged cells that join the tagged VC depart at the downstream nodes independently with probability 7/8. The tagged and untagged cells are served at the same priority level. The lost tagged PDUs are retransmitted upon negative acknowledgement messages or timeouts. The transmissions of new tagged PDUs are governed by an end-to-end flow control mechanism with PDU permits [11].

Finally, we assume that there is a cell level memory at the destination. In other words, the successful cells are stored at the receiver although the PDUs that they belong to are lost. Obviously, this feature itself has a strong impact on the overall network performance as the work left for the successive retransmission cycles decreases from one cycle to the next.

4 Simulation Results

We fixed the parameters N_A and N_P as 16 and 256, respectively, and performed various simulations to optimize M_A and M_P separately for the cases of completely partitioned and completely shared output buffers of capacity $B = 16$, 64, and 256. In these simulations, we measured the *average PDU delay* as a function of p, where p took values up to $[(1 + M_A/N_A)(1 + M_P/N_P)]^{-1}$. The average PDU delay was defined as the average time (in slots) that a tagged PDU spent in the network. In the PDU coded cases, the averages were computed over information-bearing PDUs so as to make a meaningful comparison with the uncoded case.

While optimizing M_A, we kept $M_P = 0$, and tried $M_A = 0, 1, 2, 3, 4, 6, 8$, and 12. The results of these simulations yielded $M_A = 4$ as the best choice with an acceptable throughput limitation for both output queueing techniques. In the optimization of M_P, we tried $M_P = 0, 2, 4, 6, 8, 10, 12, 14, 16, 24$, and 32, this time keeping $M_A = 0$, and obtained $M_P = 4$ and 16 as the best choices for complete partitioning and complete sharing, respectively. The simulation model and the results of these simulations for complete partitioning are discussed in detail in [12]. In this paper, we compare the results of the uncoded, only cell coded, only PDU coded, and both cell and PDU coded cases with coding parameters, M_A and M_P, chosen as above.

In Figure 1, we compare the results for complete partitioning. For $B = 16$, it is observed that PDU coding does not provide significant improvement in the delay-throughput performance for any p, and cell coding is superior to PDU coding for all p. Due to the bursty traffic characteristics, the buffers of such small capacity are almost always saturated even for low p. Therefore, cells are lost randomly with high probabilities, and consequently, cell coding outperforms PDU coding over the whole range of p. As the buffer capacities increase to $B = 64$ and 256, PDU coding starts to provide gain, and in fact for low p, yields better delay-throughput performance as compared to cell coding. For such large buffer capacities and low loads, cells are lost in rare bursts, and PDU coding exhibits gain. As the effective normalized load, $p(1 + M_P/N_P)$, approaches unity, the frequency of burst cell losses increases, and many cells from distinct connections interfere at the output buffers resulting in random cell losses. Therefore, the PDU coding gain decreases with increasing p, and cell coding starts to perform better for high p. The joint code outperforms only cell coding or only PDU coding for almost all p, except for a small degradation around $p = 0.45$ when $B = 256$, which is due to the individual performance degradation in cell coding.

In comparison of the results for complete sharing, depicted in Figure 2, with those of complete partitioning, it is observed that complete sharing yields better delay-throughput performance for low p, where the input traffic is bursty. This is in accordance with the a priori expectation that complete sharing would be effective for bursty traffic. In particular, when $B = 16$, the gain of complete sharing is quite significant, and the critical load after which retransmissions begin is shifted to about $p = 0.35$ from very small values. In fact, for low p, complete sharing with $B = 16$ achieves the performance of complete partitioning with $B = 64$. This sharing gain diminishes with increasing B.

The important difference between the results for the two queueing techniques with regard to coding is observed in the case of PDU coding when $B = 16$. Recall that, for complete partitioning with $B = 16$, cells are lost randomly with high probabilities even for low p, and hence, PDU coding provides no significant gain. Complete sharing with the same B, however, saves a significant fraction of cell losses when p is low, resulting in bursty cell loss characteristics. This, in turn, makes PDU coding effective. Comparison of the performance for the uncoded and coded cases yields similar trade-offs for both queueing techniques when $B = 64$ and 256.

In general, for low loads, cell losses occur in rare bursts, and consequently, PDU coding outperforms cell coding. For higher loads, the picture changes, and cell coding starts to perform better as cells tend to be lost randomly with high probabilities. Complete sharing, on the other hand, outperforms complete partitioning for low loads where the traffic is bursty. However, for high loads, we observe the reverse situation since the frequency of burst input cells increase resulting in a heavy load. Also, in comparison with complete partitioning, complete sharing makes PDU coding more effective especially for small B since it increases the burstiness of the cell loss process. The two coding techniques act independently, exhibiting gain and loss by means of different mechanisms. The joint code exhibits the advantages and the disadvantages of the individual contributions, in most cases with a net gain.

5 Summary and Conclusions

We have presented the results of a simulation study, showing that the use of forward error correction improves the performance of broadband networks. We have concentrated on the performance over a virtual circuit connection through an ATM network. We have considered both completely partitioned and completely shared output buffers at the switching nodes. The FEC technique is based on transmitting parity packets, which are constructed by using an erasure channel code, along with information-bearing packets. Although this may increase the network load, leading to higher packet loss rates and limit the network throughput, retransmissions are avoided provided that sufficiently many packets reach the destination. In particular, we have considered two types of coding: coding over consecutive cells and coding over consecutive fixed-length PDUs. The simulation results obtained have confirmed our a priori expectation that coding over PDUs would be effective for burst cell losses. This effect of PDU coding is comparable to that of an interleaving or a buffer management technique which can be used to combat burst cell losses. Our results indicate that, by employing FEC with correct parameters, it is possible to reduce the average PDU delays approximately to the extent of a half.

References

[1] N. F. Maxemchuk, "Dispersity routing," *Proc. ICC '75*, pp. 41.10–41.13, 1975.

[2] N. Shacham, "Packet recovery and error correction in high-speed wide-area networks," *Proc. MILCOM '89*, p. 29.5.1, May 1989.

[3] T. Kitami and I. Tokizawa, "Cell loss compensation schemes employing error correction coding for asynchronous broadband ISDN," *Proc. INFOCOM '90*, pp. 116-123, San Francisco, CA, June 1990.

[4] A. J. McAuley, "Reliable broadband communications using a burst erasure correcting code," *Proc. ACM SIGCOMM '90*, pp. 287-306, Philadelphia, PA, September 1990.

[5] E. Ayanoğlu, et al., "Protocols for error/loss recovery in broadband ISDN," *Proc. 7th Int'l. Teletraffic Congr. Sem.*, Morristown, NJ, October 1990.

[6] L. Zhang and K. W. Sarkies, "Modeling of a virtual path and its applications for forward error recovery coding schemes in ATM networks," *Proc. SICON '91*, Singapore, September 1991.

[7] E. W. Biersack, "A simulation study of forward error correction in ATM networks," *Computer Communications Review*, Vol. 22, January 1992.

[8] R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley Publishing Company, Inc., Massachusetts 1983.

[9] E. Ayanoğlu, et al. "Diversity coding: using error control for self-healing in communication networks," *Proceedings of INFOCOM'90*, pp. 95-104, San Francisco, CA, June 1990. (To be published, *IEEE Trans. Comm.*)

[10] M. G. Hluchyj and M. J. Karol, "Queueing in high-performance packet switching," *IEEE J. Select. Areas in Commun.*, Vol. 6, no. 9, pp. 1587-1597, December 1988.

[11] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 1987.

[12] N. C. Oğuz and E. Ayanoğlu, "A simulation study of forward error correction for lost packet recovery in B-ISDN/ATM," *Proc. BILCON'92*, pp. 70-80, Ankara, Turkey, July 1992.

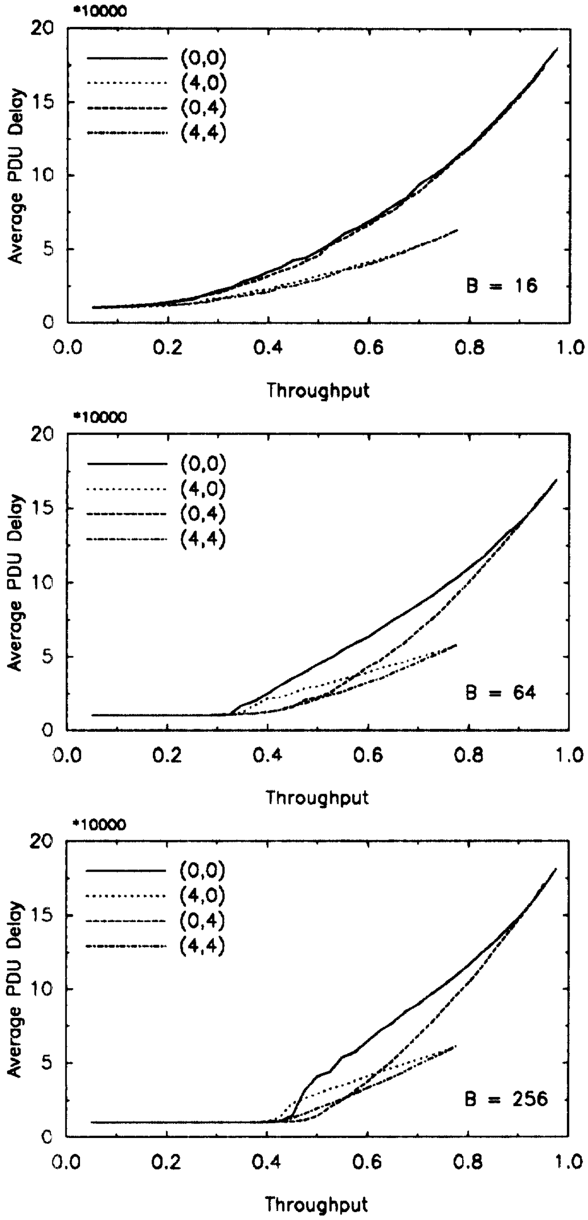

Figure 1: Comparison of the uncoded (0,0), only cell coded (4,0), only PDU coded (0,4), and both cell and PDU coded (4,4) cases for complete partitioning. Averages were computed over 512 coding blocks.

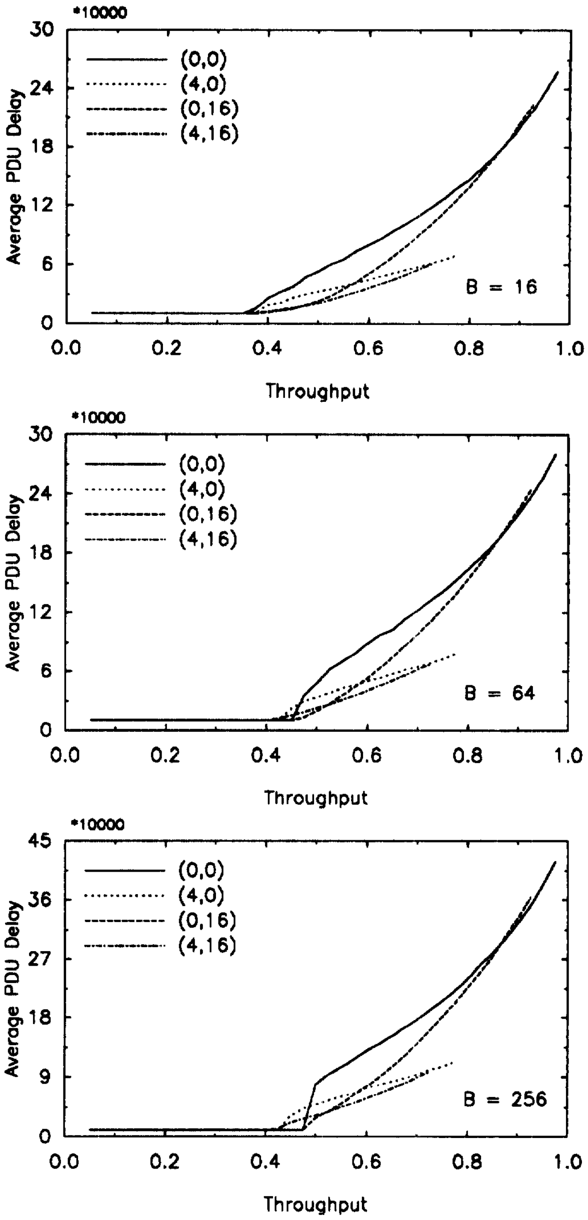

Figure 2: Comparison of the uncoded (0,0), only cell coded (4,0), only PDU coded (0,16), and both cell and PDU coded (4,16) cases for complete sharing. Averages were computed over 512 coding blocks.