

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

Nihat Cem Oğuz
Electrical and Electronics Engineering Department
Bilkent University
Ankara, 06533, Turkey

Ender Ayanoglu
AT&T Bell Laboratories
101 Crawfords Corner Road 4F-507
Holmdel, NJ 07733-3030, USA

ABSTRACT

We present the results of a simulation study for a virtual circuit connection over an ATM network where forward error correction is performed at both the ATM cell level and the packet data unit (PDU) level. A main conclusion of this study is that at low loads ATM cells from the same source dominate in the switch buffers, while at high loads there is a mixing of ATM cells from different sources. For the latter case, ATM cell level coding performs better, while for the former, PDU level coding performs better. The combination of the two techniques has the best overall performance.

1 Introduction

In broadband integrated services digital network using the asynchronous transfer mode (B-ISDN/ATM), the end-to-end propagation delays will typically be much larger than the duration of a packet. Consequently, retransmissions associated with the conventional error detection and *Automatic Repeat reQuest* (ARQ) mechanisms will increase the delay of a packet intolerably, especially for loss and delay sensitive high-speed applications. Therefore, one may use *Forward Error Correction* (FEC) to improve reliability without increasing end-to-end delay. In FEC, redundant information is sent along with the original data so that the receiver can avoid retransmissions by recovering lost information using this redundancy. However, there is a trade-off in using FEC: adding redundancy 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 *virtual circuit* (VC) connection over 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*. ATM cells are grouped into variable size *Packet Data Units* (PDUs) at the adaptation layer. Some of the PDUs arrive at the receiver with missing cells due to buffer overflows at congested nodes. By adding parity cells to each PDU, some of the lost cells can be recovered. A PDU is considered *lost* and retransmitted if its missing cells cannot be recovered by the FEC mechanism. Our principal motivation is the fact that the nature of the cell loss process strongly affects the performance of FEC. Adding parity cells to each PDU is effective when cells are lost "randomly." It is not effective when the cells are lost in bursts. In such cases, interleaving and buffer management techniques can be used. To combat burst losses, we employ FEC over PDUs, in addition to FEC over consecutive cells, and our results indicate this is effective. Unlike employing buffer management techniques in intermediate nodes, this method is simpler to implement as it does not require processing at the network nodes and can be employed selectively, e.g., only for delay and loss sensitive applications.

2 Forward Error Correction in ATM Networks

In the method of FEC over consecutive ATM cells, the encoder appends M_A independent parity cells to each group of N_A information-bearing cells. We consider this block of $N_A + M_A$ cells as a PDU. Since the receiver determines the positions of lost cells by means of sequence numbers, it is possible to design an *erasure channel* code so that up to M_A erased cells per PDU can be recovered. A lost PDU with more than M_A erased cells is retransmitted upon a time-out or a retransmission request from the receiver.

In ATM networks, once a node is congested, it remains in this state for some time resulting in consecutive cell losses. Increasing M_A is not a good solution to this problem since it leads to higher cell loss rates and limits the throughput. A better solution is to use a code over PDUs in addition to FEC over consecutive cells; each block of N_P PDUs is followed by M_P independent parity PDUs. We call this block of $N_P + M_P$ PDUs a *coding block*. This can be viewed as a two-dimensional code if consecutive cells are arranged in the form of a matrix each row of which contains cells of one PDU. The issues related to the construction of parity cells for both row and column coding are the same, and therefore, up to M_P lost PDUs per coding block can be recovered.

3 Simulation Results

In the simulations, we consider a five-hop VC connection over a wide geographical area. The end-to-end propagation delay is taken to be 10,240 slots, where a slot is defined to be the unit time needed to serve a cell at 155 Mb/sec, approximately equal to the delay of a US-wide or a Europe-wide

network at this rate. In each one of the four intermediate nodes, there is a non-blocking 8×8 ATM switch with output buffers of capacity $B = 256$ cells. To measure the coding gain, we perform FEC on the forward traffic belonging to a *tagged* source-destination pair. While passing through the network nodes, the tagged traffic interferes with the *untagged* cells belonging to other source-destination pairs. We assume that the tagged PDUs of N_A cells arrive at the source according to a Poisson process with rate p/N_A , where p is the *network load*. $N_A + M_A$ cells of a tagged PDU are transmitted in successive slots under the control of a window flow control mechanism with PDU *permits*. This mechanism prevents the network from getting into a state of ever increasing congestion since it limits the number of PDUs on the VC. The transmitter stores each tagged PDU until an acknowledgement message (ACK) is received from the destination. In the case of receiving a negative acknowledgement message (NACK), the NACK'ed PDU is retransmitted. In real life, ACKs and NACKs flow back through a similar, possibly the same, path as the forward traffic follows, and hence, are subject to loss as well as random delay. We assume for the sake of programming simplicity that they flow back through dedicated lossless, constant-delay channels. We still maintain a timeout mechanism. Similarly, we assume independent Poisson arrivals of N_A -cell untagged PDUs with rate p/N_A at the 7 untagged input ports of each node. Each untagged PDU chooses the tagged output port independently with probability $1/8$ and its cells depart from the VC at the downstream nodes independently with probability $7/8$. Also, the tagged and the untagged cells are served at the same priority level. We also assume that the receiver has a cell level memory: the successful cells of lost PDUs are stored. This feature has a strong impact on the overall network performance since it decreases the amount of work from one retransmission cycle to the next.

In the simulations, we fixed the parameters N_A and N_P as 16 and 256, respectively, and measured the *average PDU delay*, which was defined as the average time that a tagged PDU spent in the network, as a function of p . The results for the uncoded ($M_A = M_P = 0$), only cell coded ($M_A = 4$, $M_P = 0$), only PDU coded ($M_A = 0$, $M_P = 4$), and both cell and PDU coded ($M_A = M_P = 4$) cases are compared. The parameters $M_A = 4$ and $M_P = 4$ were chosen according to the results of two optimizations in which we tried $M_A \in \{0, 1, 2, 3, 4, 6, 8, 12\}$, and $M_P \in \{0, 2, 4, 6, 8, 10, 12, 14, 16, 24, 32\}$, respectively. In the PDU coded cases, the averages were computed over information-bearing PDUs so as to make a meaningful comparison with the uncoded case.

The results show the trade-off between using cell coding and PDU coding. For low p , where cells of a single connection dominate in the output buffers, losses occur in rare bursts for buffer capacities as large as 256, and hence, PDU coding outperforms cell coding. For high p , the frequency of burst losses increases, and many cells from distinct connections interfere at the switch outputs resulting in random losses. Therefore, cell coding starts to perform better as p increases. The joint code outperforms only cell coding or only PDU coding for almost all p , except for a small degradation around $p = 0.45$, which is due to the individual performance degradation in cell coding. The results of the optimizations over M_A and M_P , and the details of transmitter and receiver implementations are available from the authors.

Finally, note that if the successful cells of the lost PDUs were not stored at the destination, the average PDU delays would be much higher for high network loads. With a cell level memory at the destination, the lost PDUs have to arrive at the receiver with fewer and fewer number of cells in successive retransmission cycles. However, when there is no such memory, the PDUs are subject to the same probability of loss in successive transmission cycles.

4 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 over an ATM network. 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 ATM 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, and indicated that, by using FEC with correct parameters, it is possible to reduce the average PDU delays approximately to the extent of a half.