PERFORMANCE ANALYSIS OF A HYBRID ARQ SYSTEM IN HALF DUPLEX TRANSMISSION AT 2400 BPS

J Dalmau-Royo and J Serrat-Fernández

Universitat Politècnica de Catalunya, Spain

INTRODUCTION

Today HF communication systems rely on the concept of adaptation. In fact, the benefits of exploiting the time/frequency variant capacity of the HF channel have been reported elsewhere. Adaptive systems can play an important role in any one of the first three layers of a communication system. This paper is focused on the description of the bases of a level-2 protocol aimed to incorporate such adaptive capacity.

Hybrid ARQ/FEC protocols have been proposed to provide high data link integrities whilst keeping at the same time a high mean throughput rate. Nevertheless, hybrid ARQ strategies offer a lot of choices and none of them can be considered the optimum in any case. Among the different aspects to consider in order to choose or to design one of such protocols we must take into account the system application constraints. In this view we specify the following:

The protocol is intended to provide level-2 services in a half duplex system exploited with two different working frequencies.

Long forward messages and short replies must be allowed in order to minimize the effects of the level-1 overheads.

A residual bit error rate of about 10^{-4} can be tolerated at rough bit rates of 2400 bps.

Although we will go insight throughout the paper it is enough to justify some of the approaches that we have considered.

In the following section we present the alternatives which were considered. Then we proceed describing the three which apparently were the most promising candidates. Finally we discuss the laboratory tests results with emphasis on the channel models used for the tests.

PROTOCOL AND CODING STRATEGIES

The HF channel introduces a mixture of random and burst errors, the so called diffuse errors as described by Darnell and Tech [1]. As a consequence the coding strategies should be suitable to combat this type of errors.

The basic diffuse error correcting techniques are the following ones:

- Error detection and retransmission
- Simple interleaving
- Fire codes
- Reed Solomon codes
- Threshold decoding techniques
- Code combining

Error detection and retransmission is generally easy to implement. Moreover, it requires less redundancy than others to correct the same number of errors and it is independent of the burst length. However, this technique requires a reliable feedback link and enough buffer storage and logic at both the transmitter and the receiver. In addition it presents low throughput rates in noisy forward channels.

Simple interleaving combats the effects of error bursts by distributing them over separate coded data streams. It is also easy to implement and many good random error correcting codes are known. Its main drawback is that the errors correlation is not fully exploited in decoding. Moreover, interleaving may still be inadequated with very long bursts.

Fire codes present a very high efficiency and are of simple implementation. Yet, long Fire codes are required for a moderated burst correction capability.

Although Reed Solomon codes show a code efficiency typically greater than 90%, one of its main drawbacks is concerned with their rigid correction capability, as it has been stated by Reed and Hopkinson [2].

Threshold decoding techniques are able to handle large bursts and they are also commercially available. However, the code design is still an art and devices exist for only a limited range of code rates.

Code combining presents an arbitrarily low output error rate even under the most severe channel conditions. The strategy is the code rate adaptation to the channel conditions. However, it presents a very low throughput under severe conditions.

Once the basic diffuse error correcting techniques have been examined, it can be concluded that a suitable blend of them should be the best solution to cope with such error patterns. In this way, a strategy based on the following premises was pursued:

- Error detection and correction by retransmission
- Use of simple interleaving
- A certain degree of code rate adaptation to the prevailing channel conditions

These considerations lend to a type-II hybrid ARQ, as in reference [1].

The design of such coding strategy must consider the following aspects:

- The selection of codes to bring out an optimal process of error detection and correction.

- The interleaving parameters.

- The way to adapt the code rate.

- The error protection of the feedback channel.

Because of its half duplex operation, the selection of the retransmission strategy (Go Back-N, Selective Repeat, etc.) is clearly overcome. In addition, the interleaving depth is constrained and limited by the frame length.

BCH and RS are known as ones of the most efficient codes in terms of trade off between error protection capability and rate. Because of this, we choosed BCH codes since the begining although without discarting RS codes. In order to provide a reasonably high error protection whilst keeping at the same time a short code block length, we picked up code lengths of 15 and 31 bits. Specifically we selected the BCH(15,10), BCH(15,5) and BCH(31,11) in order to be used as explained in the following section.

As far as the feedback channel is concerned we are limited by the short length of the feedback frame. Then, codes with diffuse error correction capability must be adopted here. For instance, diffuse convolutional codes are a valid alternative. This is because we selected this family of threshold decodable codes to cope with the error patterns of this channel and specifically a convolutional code of length 14.

PROPOSED APPROACHES

In the following paragraphs we specify three protocol strategies which are from now identified as System 1, System 2 and System 3. The three systems exchange forward frames, conveying the information from the transmitter to the receiver, and backward frames only for acknowledgement purposes. The information bits to be transmitted are segmented in blocks, encoded and placed, totally or partially, into the transmission buffer with other blocks in order to constitute the forward frame. Each block is identified by its position in the frame. If the receiver is unable to decode or if it detects errors into a given block a NACK is sent back in the position associated to this block in the backward frame. After the receipt of a NACK the transmitter may encode again the block with a different code (Systems 1 and 2) or it may send the remaining bits of the first encoding process (System 3) in order to allow the receiver to do a reliable error correction process. Therefore, the three systems may use only one or more forward frames to insure that each segment of information bits is appropriately delivered at the distant end.

System 1

System 1 makes use of segments of 10 information bits. The first time that a segment is picked up it is encoded with the BCH(15,10). In this case the receiver exploits the properties of the BCH(15,10) to correct one error or to detect two errors in the block. Therefore only when two errors are detected, a NACK will be sent back and the receiver will be waiting for the retransmission of this segment. On the other cases, the 10 information bits will be delivered to the user after the appropriate error correction process if necessary.

When a NACK is received, the transmitter splits the segment in two halves, encodes independently each one with the BCH(15,5) and places both blocks into the next forward frame. Now the receiver will exploit the capacity of this code to correct up to three errors or to detect if the decoded block is really a codeword. In the event that the decoded block is not a codeword, a new NACK would be sent back in order to force the transmitter to forward again the same code block.

Summarizing, since the first retransmission each half segment is encoded with a BCH(15,5), transmitted as many times as necessary and processed independently of the other half. The number of retransmissions is limited to a reasonable amount and appropriate control procedures are carried out in order to always identify the position of a given half segment of information bits inside the forward frame.

The ACK/NACK bits of each received block are encoded with a half rate convolutional diffuse code to constitute the backward frame.

System 2

System 2 exhibits only a sligth difference with respect to System 1 concerning the way in which it exploits the BCH(15,10) code. This is, the BCH(15,10) is now used only to detect but not to correct errors in the first transmission of a segment. Therefore a NACK will be sent back to the transmitter if 1,2,3,5,6, .. errors appear inside the 15 bit block. Besides this, System 2 proceeds in the same way as System 1.

System 3

System 3 works with segments of 11 information bits encoded with the BCH(31,11). Among the 31 bits at the encoder output only 16 are selected and stored in the transmission buffer to be included in the next forward frame. These are the 11 bits of the segment and 5 redundancy bits specially selected in order to maximize the minimum distance between the 16 bit codewords. By computer search we realized that the minimum free distance is 3. This means that the receiver will be able to detect up to 2 errors in each 16 bit block.

When some error is detected a NACK will be sent back. Then, after the receipt of a NACK the transmitter will load the remaining 15 redundancy bits plus a stuffing bit in the transmission buffer in order to be forwarded in the next frame. Therefore the receiver will be enabled to decode the 31 bit block of the BCH(31,11) transmitted in two consecutive forward frames. The framework of this procedure is explained by Kallel [3] and it can also be seen as a modified memory ARQ as described by Serinken [4].

In case that the first transmitted block were considered error free, the receiver would deliver 11 information bits to the user and an ACK would be issued. Then the transmitter would delete the above mentioned

15 redundancy bits and it would use this position in the buffer to send a new encoded segment.

The backward frame is built up from the ACK/NACK bits of each received block as in Systems 1 and 2.

LABORATORY TEST BED

In order to assess the performance of the three systems several tests were carried out based on the arrangement shown in Figure 1. In this figure PC1 emulates a transmitting station and PC2 the receiving station. The link between both stations is a three wire line plugged on the asynchronous serial port of each PC.

The processing involved in each of the three systems is carried out by an ad-hoc software developed for each PC. On the other hand, as no modem or channel simulator are used, the effects of the HF channel need to be simulated at one or both PCs. In order to do it a total of nine channel models comming from the same number of real HF links were considered. Because of the involvement of the results and their significance with the used models we devot a part of the present section to deal with this topic.

Figure 2 shows the functional block diagram of the software running on the transmitter (PC1). On top of its layered structure we have the module devoted to the user interface which is intended to allow him the programming and initialization before its execution. Among the initialization parameters the user must specify the name of the file to be transmitted, the type of protocol and the HF simulation model. At the bottom we have the communication module which for these tests was designed to transmit/receive characters through the asynchronous port of the PC. The interface between the communication module and the upper layers has been carefully specified in order to allow other communication links in the future. The other layers constitute the core of the ARQ protocols described in the previous section in order to process the backward frames and to generate the forward frames. Concerning the software of PC2 it looks like Figure 2 but now the intermedium levels are intended to process the forward frame and to generate the backward frame. An special feature of the user interface at PC2 is that it also generates a report file containing the most relevant data of the transmission/reception process like the number of residual errors, mean throughput rate, number of blocks transmitted one time two times, etc. All the results presented in the next section have been obtained from the report files written at the end of each run.

HF channel models

The HF channel is based on a discrete model derived from two sets of experiments:

- HF Ionospheric communications at 2400 bps with PSK-2 modems reported by Spunticchia and Caroggio [5].

- HF ground wave mobile communications at 600 bps with FSK-2 modems reported by Dalmau [6].

The second set of experiments corresponds to data obtained in the metropolitan area of Barcelona. Due to the short length of the links (between 5 and 30 Km) the main propagation mode is ground wave. The registered error patterns exhibit diffuse errors like those encountered in sky wave propagation. Therefore, the results got from tests using these channels may be also extensive to any HF channel.

A Fritchman's [7] partitioned three state model was selected because it gives the best trade off between accuracy and complexity.

TEST RESULTS

In Figure 3 to 5 we show a summary of the results obtained after several runs. Each run was intended to transmit a file of 60 Kbits from PC1 to PC2 in order to evaluate the user's bit error rate and the mean throughput rate.

The user's BER is computed dividing the residual errors in the received file by 6 10^4. It can be argued that this procedure does not give a reliable estimate mainly if the error count is low. This is because we write in the appropriate figures the absolute error count.

As far as the mean throughput is concerned it must be said that it is computed dividing the 6 10^4 information bits by the total number of bits transmitted through the channel in both directions but excluding the start and stop bits of each character in the asynchronous link. Then it gives an estimate of the maximum attainable mean throughput rate of a hypothetical synchronous system using the same rough bit rate for both modems.

Figure 3 plots user's BER versus channel BER for the nine channels and the three systems considered. The number of blocks per frame is always 160 and the backward frame is encoded by the 1/2 rate diffuse convolutional code. As it can be seen the three systems show the same behaviour up to channel BERs of about 4 10^{-3}. At this point System 1 begins to lose performance with respect to the others. On the other hand, System 3 falls between 1 and 2 except for the worst channel where it delivers up to 11 errors. The same behaviour was also observed in an ideal backward channel.

This results are consistent. In fact, System 1 loses error detection capability compared with System 2 because the BCH(15,10) is also used to correct errors. Also System 3 is less powerful as far as error detection is concerned because the minimum distance of its error detecting code is 3 instead of 4.

Figure 4 shows the mean throughput rate of the three systems versus channel BER. Again the number of blocks in the forward frame is 160 and the backward frame is encoded by the same convolutional code. System 3 clearly achieves the highest throughput because of its particular phylosofy as a modified memory ARQ system, thus matching better the code to the channel. No significative differences were observed considering an ideal backward channel.

Finally, figure 5 is a plot of the user's errors versus the frame length for System 3 working in the worst channel (BER = 10^{-2}). The number of errors depends on the frame length as it could be expected because this one is affecting the error pattern distribution. Excluding the shortest frame case, the mean number of errors is 3 over the 60000 information bits, a good enough figure in respect of the design objectives. No differences were observed as far as throughput nor considering an ideal backward channel.

CONCLUDING REMARKS

System 2 clearly outperforms System 1 considering user's errors and it shows a low throughput loss with respect to it.

System 3 achieves the highest throughput for all the channels considered. In fact, the difference with respect to Systems 1 and 2 is considerable. As fas as user's errors is concerned, although it is outperformed by System 2 for channel BERs higher than 6.5 10^{-3}, it seems to be the best choice assuming a long enough forward frame. On the other hand, its implementation complexity is also lower than the other systems.

The protection of the backward channel is enough to cope with the errors introduced by all the channels considered. Actually, the fact of considering the back channel ideal or not does not affect the results.

REFERENCES

1. Darnell, M. 1985, "Problems of Mobile HF Communication and Techniques for Performance Improvement". IEE Proceedings, 5, August.

2. Reed, A.P.C. and Hopkinson, J.N., 1990, "Adaptive Data Link Protocols: Design and Performance over Automated HF Skywave Links". Plessey, U.K.

3. Kallel, S., 1990, "Analysis of Type II Hybrid ARQ Scheme with Code Combining". IEEE Trans. on Comms., 38, 8, August.

4. Serinken, N.M., 1988, "HF Data Message and Facsimile Terminal System". 3th IEE HF Systems and Tech., London.

5. Spunticchia and Caroggio, 1986, "Un Procedimento di Misura e di Analisi degli Errori per Sistemi di Transmisioni Dati". XVII Congresso Int. per l'Elettronica".

6. Dalmau, J.L., 1991, "A Contribution to the Analysis of Coding Schemes for Mobile Data Communications over HF Channels". Doctorate Thesis, Universitat Politècnica de Catalunya, Spain.

7. Fritchman, B.D., 1967, "A Binary Characterization using Partitioned Markov Chains". IEEE Trans. on Inf. Theory, IT-13, 2, April.

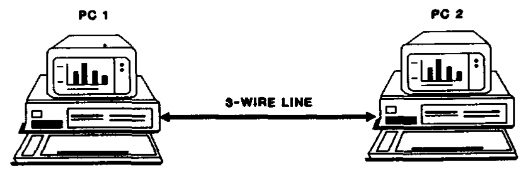

FIGURE 1: LABORATORY TEST BED

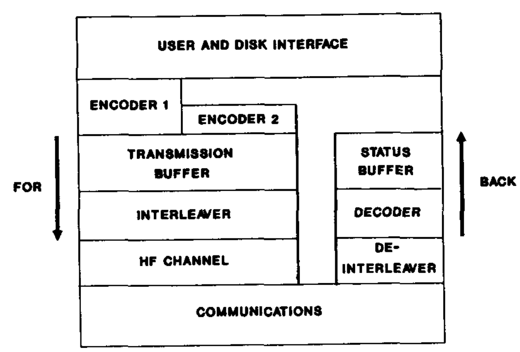

FIGURE 2: LAYERED STRUCTURE OF THE SOFTWARE RUNNING IN PC 1

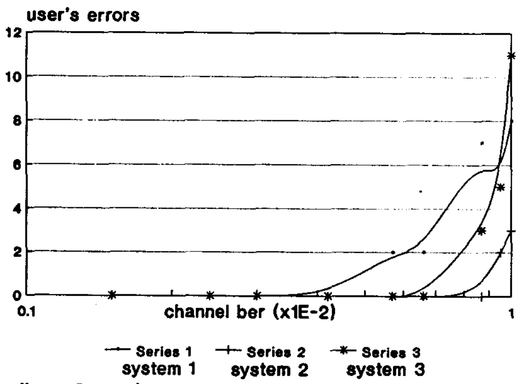

figure 3: user's errors vs channel ber

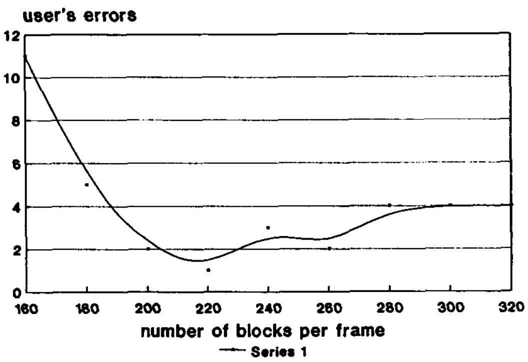

figure 5: user's errors vs number of blocks per frame

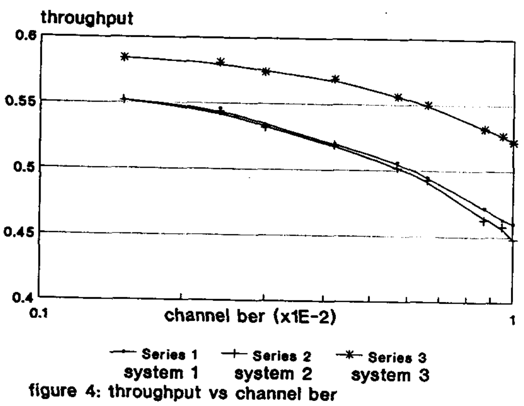

figure 4: throughput vs channel ber