Robustness of Multiplexing Protocols for Audio-Visual Services over Wireless Networks *

W. S. Lee, M. R. Frater, M. R. Pickering and J. F. Arnold
School of Electrical Engineering
University College UNSW
Australian Defence Force Academy
Canberra, ACT 2600.

Abstract

The protocols used to multiplex the various streams in an audio-visual service (such as MPEG 2 Systems and ITU-T H.223) have an important impact on the quality of the service. The fact that wireless channels are often associated with high bit-error rates which are often bursty means that the multiplexing protocols should be designed to be robust against such errors. In this paper, we suggest two departures from traditional practice which can improve significantly the performance of a packet-oriented multiplex for use with audio visual services: 1) where the packet header is protected against errors using forward error correcting codes, the use of a synchronization codeword (or flag) is often unnecessary, and 2) for channels where errors tend to occur in bursts, the occurence of errors in the header can be reduced by transmitting the header information twice (at both ends of the packet) with a low level of FEC protection instead of using the same number of bits to obtain an increased level of protection using a single code word. The value of these approaches is confirmed by simulations.

1 Introduction

In the transmission of audio-visual services, the loss of information in the multiplexing layer often has a significant effect on the quality of the service (e.g. see [7] for a discussion on the effects of losses in the multiplexing layer on video quality). The ability of multiplexing protocols to cope with the high error rates that occur in mobile networks, such as the DECT (Digital European Cordless Telephone) system, is hence an important issue. To be suitable for use over such networks, a multiplexing protocol should be robust against two types of error: bit errors, which will often occur in long bursts, and bit insertions and deletions, which are observed in some cellular systems where a handoff occurs.

Most multiplexing protocols currently used for audio-visual services (such as H.223 [5] and MPEG 2 Systems [3]) take the form of a two layer packet-based multiplex. In the upper layer, often called the "adaptation layer", data from each of the elementary streams (e. g. a single channel of video or audio) is packetized. In the lower layer, the packets from all streams associated with a call are time-division multiplexed onto a single channel.

In general, a packet consists of four parts: a synchronization marker or flag, to enable the receiver to unambiguously identify the location of the start of a packet, a header, usually containing address and other information, a body, containing the information to be sent, and a trailer. The robustness is sometimes improved by the inclusion of error control codes (such as a CRC or BCH code [6]) in one or more fields. In some cases, one or more of these fields may be specified to be empty.

In this paper, it is shown that the performance of multiplexing protocols can be improved by adapting two techniques in addition to those normally used. If the multiplexer packet header is protected against errors using forward error correction (FEC) codes, the use of a synchronization code word (or flag) is often unnecessary. Removal of this component of the packet can lead to significant reductions in overhead. For channels where errors tend to occur in bursts, the occurence of errors in the header can be reduced by transmitting the header information twice (at both ends of the packet) with a low level of FEC protection instead of using the same number of bits to obtain an increased level of protection using a single code word.

2 Packet Loss

There are two ways in which all the data in a multiplexer packet may be lost to the receiver. First, incorrect operation of the synchronization process may cause unsynchronized data to be passed on to the receiver or correctly synchronized data to be discarded because the receiver believes incorrectly that it has lost synchronization. Secondly, if the header of the packet is corrupted, the receiver will be unable to correctly decide the content of the packet. The first problem is usually handled by the addition of a synchronization flag with the packet while the second problem can be handled using error correcting codes.

2.1 Packet Loss Due to Synchronization Errors

The synchronization marker of a packet is usually a codeword with good autocorrelation properties such as a Barker code or a pseudo noise sequence [2]. The first step in the procedure for maintaining synchronization is to compare the correct codeword with the received synchronization markers of the next k_1 packets. If the total number of bit errors between the correct codeword and the received markers is greater than some threshold (denoted by T_1), then it is decided that synchronization has been lost and the regain synchronization function is performed. Regaining synchronization is performed by stepping through the incoming bitstream one bit at a time looking for a set of markers in the next k_2 packets for which the total number of bit errors when compared to the correct codeword is less than some threshold (denoted by T_2).

To estimate the performance of the synchronization marker, we use the following three measures.

In the maintaining synchronization function, channel errors in the synchronization markers may cause the function to falsely decide that synchronization has been lost. As a result one or more packets of correctly synchronized data will be discarded while synchronization is regained unnecessarily. The probability of this occuring is P(deciding not in sync|in sync) and is denoted by PDNS_S.

In the maintaining synchronization function, the algorithm may fail to detect that synchronization has been lost because of a bit insertion or deletion. This failure will cause a packet of unsynchronized data to be passed on to the receiver. The probability of this occuring is P(deciding in sync|not in sync) and is denoted by PDS_NS.

In the regaining synchronization function, a set of k_2 groups of bits may be similar enough to the correct codeword for the function to falsely decide that synchronization has been regained. Alternatively, if the correct set of k_2 markers is compared with the codeword, the function may falsely decide that synchronization has not been regained. The probabilty of the regaining synchronization function failing to correctly regain synchronization at the next available set of k markers is P(not regaining sync) and is denoted by PNS.

For the experiments we conducted, it was assumed that the probability of losing synchronization because of a bit insertion or deletion is negligible. However, in order to choose a realistic value for T_1 we have chosen a maximum value for PDS_NS of 0.2. Hence T_1 and k_1 were chosen to give a satisfactory value for PDNS_S while still satisfying this criterion. T_2 and k_2 were chosen as the smallest values which gave a satisfactory value for PNS.

3 Packet Configuration

In this section, we describe the configuration of a multiplexer packet which sends the FEC protected header information twice and does not use a synchronization marker. For comparison , we also describe two other more conventional configurations.

In order to increase the robustness of the multiplexing protocols for transmission over wireless networks, we will consider configurations with fixed packet lengths.

The new packet configuration called DOUBLE has no synchronization marker and an FEC protected header that is sent twice, once at the begining of the packet and once at the end of the packet. For this configuration the synchronization functions use the following simple strategy: Match the first and second code words in the packet and compare the number of mismatched bits to the threshold values T_1 and T_2 to decide whether the packets are properly synchronized.

This configuration is compared to the following two more conventional configurations:

BLOCK The packet starts with a synchronization flag and a header protected by error correcting codes. This is followed by the information bits.

INTERLEAVE The packet starts with a synchronization flag. The FEC protected header is interleaved throughout the information bits so that they are evenly spread throughout the packet.

The BLOCK configuration is a simple protocol similar to H.223 (except for fixed packet length). The INTERLEAVE configuration is used in the H.22P/M [4] proposal for low bitrate multimedia mobile communication.

We conducted the experiments described in the following section to determine the packet loss performance of the three packet configurations for several different channel error characteristics.

4 Experimental Procedure

The experiments were conducted for four channel models:

1. a DECT channel with bit error rate 2×10^{-2} (DECT1) [1],
2. a DECT channel with bit error rate 2×10^{-3} (DECT2) [1],
3. a binary symmetric channel with bit error rate 2×10^{-2} (BSC1) and
4. a binary symmetric channel with bit error rate 2×10^{-3} (BSC2).

The errors in DECT1 and DECT2 are extremely bursty with the bursts in DECT1 much longer and more frequent than the bursts in DECT2.

For the BLOCK and INTERLEAVE structures we used the 31 bit pseudo-noise sequence "1111100011011101010000100101100" as the synchronization marker and a (31,6) BCH code to protect the header. The same pseudo-noise sequence is also used in the H.22P/M proposal. For the DOUBLE structure we tested two different configurations. For the first configuration we used two copies of a (31,6) BCH protected header at both ends of the packet and in the second configuration we used a (15,7) BCH code. The plots for PDNS_S, PDS_NS and PNS versus threshold for $k = 1$ and $k = 4$ are shown in Figure 1.

4.1 Determining values for T and k

The values of T_1, T_2, k_1 and k_2 were determined for each strategy by first determining the values of PDNS_S, PDS_NS and PNS versus threshold for several different values of k. Figure 1 shows a plot of these values versus threshold when $k = 1$ and $k = 4$ for the DECT1 channel using the DOUBLE configuration with a packet length of 1024 and a (15,7) BCH code.

Packet Size	*Channel*	*Strategy*	*PH*	*PS*	*PT*
1024	DECT1	BLOCK	2.1×10^{-2}	$< 1 \times 10^{-5}$	2.1×10^{-2}
		INTERLEAVE	1.6×10^{-2}	$< 1 \times 10^{-5}$	1.6×10^{-2}
		DOUBLE	7.9×10^{-3}	9.9×10^{-4}	8.6×10^{-3}
	DECT2	BLOCK	1.9×10^{-3}	$< 1 \times 10^{-5}$	1.9×10^{-3}
		INTERLEAVE	5.4×10^{-4}	$< 1 \times 10^{-5}$	5.4×10^{-4}
		DOUBLE	2×10^{-5}	$< 1 \times 10^{-5}$	2×10^{-5}
	BSC1&BSC2	ALL	$< 1 \times 10^{-5}$	$< 1 \times 10^{-5}$	$< 1 \times 10^{-5}$
512	DECT1	BLOCK	2.1×10^{-2}	$< 1 \times 10^{-5}$	2.1×10^{-2}
		INTERLEAVE	2.0×10^{-2}	$< 1 \times 10^{-5}$	2.0×10^{-2}
		DOUBLE	1.5×10^{-2}	5.3×10^{-3}	1.9×10^{-2}
	DECT2	BLOCK	2.0×10^{-3}	$< 1 \times 10^{-5}$	2.0×10^{-3}
		INTERLEAVE	1.2×10^{-3}	$< 1 \times 10^{-5}$	1.2×10^{-3}
		DOUBLE	4.8×10^{-4}	$< 1 \times 10^{-5}$	4.8×10^{-4}

Table 1: Performance of the multiplexer packet configurations. PH is the probability of packet loss from corrupted headers. PS is the probability of packet loss from synchronization errors. PT is the probability of packet loss from both corrupted headers and synchronization errors (this may be less than the sum of the two losses because some packets lost by synchronization errors are also lost because of header errors).

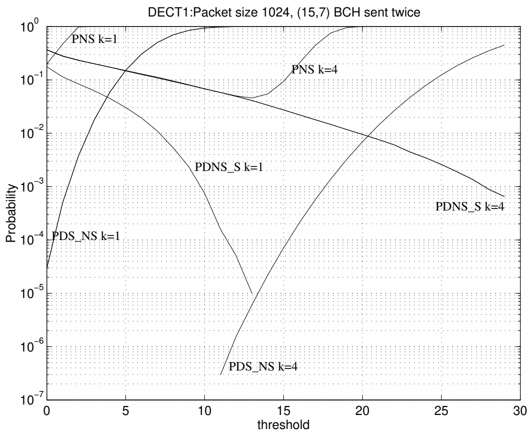

Figure 1: Plots for synchronization experiments using DOUBLE structure. DECT1, packet size 1024.

The values for PDS_NS were calculated by taking the average number of times that the maintaining synchronization falsely assumed synchronization for every positions in a sample corrupted bitstream except the correct position. We calculated PNS by running the regaining synchronization function from the first position in the packet after the correct synchronization position up to the next correct synchronization position and checking if we have acquired synchronization correctly.

For the case shown in Figure 1, k_1 was chosen as 4 and T_1 was chosen as 26. It can be seen from Figure 1 that these values guarantee PDS_NS is less than 0.2 and PDNS_S is suitably small with a value of approximately 2.5 x 10^{-3}. Then k_2 was chosen as 1 and T_2 was chosen as 0 since these values give a PNS value of approximately 0.2. This value of PNS is satisfactory since the probablity of the regaining synchronization function falsely being called is PDNS_S 2.5 x 10^{-3}. The results in Section 5 show that the values of T and k are not critical to the performance of the system because the packet losses are predominantly caused by header errors.

4.2 Determining Packet Loss

The following simple strategy for determining packet loss due to synchronization is used. When a synchronization loss is detected (falsely in our experiments since we do not consider actual synchronization loss), the receiver will consider the next packet as a lost packet and call the function to regain synchronization. For our measurements, all the packets until synchronization is correctly regained are considered lost. (This includes the packets lost because the receiver acquired synchronization at the incorrect place. In such a case, the packets are lost until the error is detected again and synchronization is acquired at the correct place.)

4.3 Decoding the Header

For the DOUBLE configuration, decoding of the header information is done by first detecting the errors in the two code words. The code word with the smaller number of detected errors is assumed to give the correct header.

4.4 Decoding Delay

One of the disadvantages of using the DOUBLE and INTERLEAVE configurations is the delay caused by having to receive the whole packet before decoding can begin. By using the DOUBLE strategy, delay can be avoided when the channel is not in error, by accepting a slight reduction in performance. This is achieved by assuming that the first word is correct if no error is detected. If an error is detected in the first word, then the code word with the smaller detected error is used. In this way, delay happens only when an error is detected. This decoding strategy can be extended so that the second word is used only when more than one (or any number) of errors are detected in the first word.

Packet Size	*Channel*	*PH*	*PS*	*PT*
1024	DECT1	1.8×10^{-2}	3.1×10^{-3}	2.0×10^{-2}
	DECT2	1.9×10^{-4}	$< 1 \times 10^{-5}$	1.9×10^{-4}
	BSC1	4.4×10^{-5}	$< 1 \times 10^{-5}$	4.4×10^{-5}
	BSC2	$< 1 \times 10^{-5}$	$< 1 \times 10^{-5}$	$< 1 \times 10^{-5}$
512	DECT1	2.9×10^{-2}	1.0×10^{-2}	3.6×10^{-2}
	DECT2	1.2×10^{-3}	$< 1.5 \times 10^{-5}$	1.2×10^{-3}

Table 2: Performance of Double structure with (15,7) BCH code word.

4.5 Unequal Interleaving of the Codeword

One potential disadvantage of using the DOUBLE structure, as opposed to using a synchronization word with good autocorrelation properties such as a Barker code, is that the number of mismatched bits will be small if the sequence has only shifted a small amount (such as from an insertion or deletion error). This means that P(deciding in sync|not in sync) will be high for bit insertions or deletions and the probability of acquiring synchronization incorrectly is also high. We mitigate this effect by using unequally spaced interleaving for the code words. Note that equally spaced interleaving will still suffer badly from increased probability of acquiring synchronization incorrectly. Since we would like the two code words to be as far away temporally as possible, it is important that the interleaved code words are interleaved over as small a region as possible. We use a simple scheme where the number of bits between the nth bit and the $n+1$st bit is n for the first code word. To concentrate most of the bits of the second code word near the end of the packet we use an interleaving scheme which is the mirror image of the first scheme, i.e. the nth bit and the $n+1$st bit are separated by $code\ size - n$ bits. (For the (31,6) BCH code words, a slightly different scheme, where the difference between the nth and $n+1$st bit is approximately $n/4$ is used so that the code words are concentrated near the ends of the packets.)

Note that it is possible to use the code words for synchronization purposes even if only one code word is sent, such as in the BLOCK and INTERLEAVE structures. In this case, we can use the number of errors detected as information for deciding whether we are in synchronization. If cyclic code words such as the BCH code are used, it is also advisable to use some unequal spaced interleaving for the same reason that it is used in the DOUBLE structure (in cyclic codes, a cyclic shift of a code word is another code word).

5 Results

The results of the packet loss experiments are shown in Table 1. These results show that, when using the same number of bits for synchronization and header protection, the DOUBLE structure performs better for bursty channels such as DECT1 and DECT2. This is mainly because the DOUBLE structure performs much better in combatting header loss when the channels are bursty even though its synchronization performance is slightly poorer.

Table 2 shows the results when the DOUBLE structure is used with two (15,7) BCH codes instead of two (31,6) BCH codes. The performance of this DOUBLE structure is comparable to that of the INTERLEAVE structure even though it uses only half as many bits.

It is important to compare the results of sending a single (31,6) BCH protected header interleaved throughout the packet with sending two (15,7) BCH protected headers. The results show that for the binary symmetric channel, sending one longer code word is better than sending two smaller ones as we would expect. However, for the DECT channels, the results show that, for a long enough packet, sending two shorter code words at the begining and end of the packet may give a better result than sending one longer code word. The technique is particularly effective for the DECT2 channel at packet length 1024. A plausible explanation for this is that most of the bursts are shorter than the packet length and far enough apart that the two code words at opposite ends of the packets are rarely affected by error bursts at the same time. In such a situation, the decoding error from taking the code word with the smaller detected error of the two code words will be small. In contrast, each error burst will still corrupt a significant portion of at least one interleaved code word if the interleaving (which is restricted by the packet size) is not deep enough.

6 Conclusion

We have shown through simulation that, when a fixed packet length is used, a synchronization marker is often unnecessary if error correcting codes are used to protect the header. Furthermore, for bursty channels, the occurence of errors in the header can be reduced by transmitting two copies of the header with a low level of FEC protection instead of protecting one copy of the header with a longer FEC code. Although the simulations were done only for the DECT channels, we expect similar results for other bursty channels.

Acknowledgements

This research was supported by the Australian Research Council.

References

[1] P. Crespo, R. Mann Pelz, and J. Cosmas. Channel error profiles for DECT. *IEE Proceedings-Comunications*, 141(6):413–420, 1994.

[2] J. E. Hershey and R. K. R. Yarlagadda. *Data transportation and protection*. Plenum Press, New York and London, 1986.

[3] ISO/IEC International Standard 13818-1. *Generic coding of moving pictures and associated audio information: Systems*. 1995.

[4] ITU-T Draft Recommendation H.22P/M. *Multiplexing Protocol for Low Bitrate Multimedia Mobile Communication*.

[5] ITU-T Recommendation H.223. *Multiplexing Protocol for Low Bitrate Multimedia Communication*.

[6] S. Lin and D. J. Costello. *Error control coding*. Prentice-Hall, Englewood, NJ, 1983.

[7] J. Zhang, M. R. Frater, J. F. Arnold, and T. M. Percival. MPEG 2 video services for wireless ATM networks. *IEEE Journal on Selected Areas in Communications*. To appear.