

Method for correcting errors in packet data

Prior art

5 For data transmission in packet form, the DAB (Digital Audio Broadcasting) standard, for example, specifies a method for transmitting data in packet mode. This method provides for an error recognition method for recognizing data packets transmitted incorrectly (CRC
10 method). Since the individual packets are combined to form data groups and also to form data files, a single incorrect bit can ruin the transmission of large volumes of data. Error protection for DAB is achieved by means of a convolution code. The objective in this
15 context was to achieve a bit error rate of 10^{-4} . However, practical experience has shown that although the bit error rate is sufficiently low for audio, it is too high for transmitting data services.

20 In the DVB-T method, a concatenated code comprising convolution code and block code is used. This allows very low bit error rates to be achieved in the order of magnitude of 10^{-8} .

25 Advantages of the invention

With the measures in claim 1, where code words are used over a plurality of data packets, redundancy information for correcting errors is added while
30 retaining the original packet data structure, specifically at the cost of a free data field or of a stuffing data field or useful data field, and where the cycle of error protection, particularly its code word length, is chosen as a multiple of a minimum size for a
35 packet length, errors can be corrected without a DAB receiver, for example, needing to have knowledge about this method, nevertheless recognizing the data as valid DAB packet mode data and being able to process them using an ordinary error recognition method (CRC).

Higher-quality receivers with control over the error correction method based on the invention can use this method and thus deliver correct data files more often or more quickly.

5

Drawings

Exemplary embodiments of the invention are explained in more detail with reference to the drawings, in which,
10 figure 1 shows a structure for data packets on the basis of the DAB packet mode,
figure 2 shows the insertion of redundancy information in the padding field of the packets,
figure 3 shows the arrangement of packets from figure 2
15 in a cohesive data stream,
figure 4 shows the insertion of redundancy information into a single packet without useful data,
figure 5 shows the arrangement of packet data from figure 4 in a cohesive data stream,
20 figure 6 shows an improved structure for the packets for distributing burst errors.

Description of exemplary embodiments

25 Figure 1 shows the structure of a data packet on the basis of the DAB packet mode. The packet header accommodates particulars regarding the packet length, a continuity index, first/last, an address field, a command field and particulars regarding the useful data
30 length. This packet header is followed by the packet data field containing the useful data (useful data field) and the field for padding data, e.g. stuffing data. There then follows a 16-bit field for error recognition (packet CRC).

35

Valid lengths for packets are 24, 48, 72 or 96 bytes. This structure needs to be maintained for the inventive solution in order to allow conventional receivers to perform decoding.

Error correction can be performed using any method which uses a systematic code. That is to say that the data to be protected are maintained unchanged and only
5 redundancy information in the form of additional bytes is additionally added and used for error correction. An example which will be used is a REED-Solomon code RS (255, 239), which is shortened to the packet sizes used. Code shortening is a known method for these
10 codes.

The invention proposes two embodiments for inserting the redundancy information:

1. Insertion of the redundancy information in the
15 padding field of the packets. A normal packet mode decoder will rate it as padding and ignore it.
2. Insertion of the redundancy information into a single packet without additional useful data, so that the normal packet mode decoder can simply
20 ignore this packet.

Figure 2 shows the structure for the first alternative. The packet header requires 3 bytes. There then comes a useful data field of 0 to 74 bytes, followed by a
25 padding field, the SYNC field, the redundancy information with 16 bytes and 2 bytes for error recognition (CRC).

The SYNC byte is used for recognizing packets which
30 contain redundancy bytes. Alternatively, the command bit (bit 7) from the packet header can be set in such packets.

The arrangement of the packets shown in figure 1 in a
35 cohesive data stream is shown in figure 3. In this example, the code is shortened to the code word length (192/176). The first value, 192, in this case is the total code word length of the REED-Solomon code, and the second value 176, is the length of the useful data

to be protected. The calculation of the redundancy bytes of the code word does not contain the CRC for the current packet with redundancy bytes, but rather that for the previous packet with the redundancy bytes. The
5 CRC for the current packet is then calculated using the redundancy bytes. The code word length is therefore both 192 and the sum of the packet lengths. In this case, the packet header is denoted by PH, S is the sync data and R is the redundancy information for correcting
10 errors.

In this embodiment, the cycle for the error protection matches that for the packets. That is to say that the code word length of the additional error protection
15 needs to be a multiple of the minimum size for a packet length, in this case 24. Hence, by way of example, code word lengths of 168, 192, 216 and 240 are also conceivable, besides others. The chosen code word length is then filled with an integer combination of
20 the four possible packet sizes.

To be able to calculate the CRC for the packet containing the additional redundancy information, the CRC bytes in the next respective code word are handled.
25 If they were to be situated in the same code word, the CRC calculation would not be possible, since the redundancy bytes of the RS code would also need to be calculated with the CRC bytes, but these in turn cannot be calculated until the RS bytes are definite. Hence,
30 the RS bytes are therefore calculated first and inserted into the packet. Next, the CRC bytes are calculated and appended. The fact that they are not in the same RS code word does not alter the RS bytes.

35 The second alternative for inserting the redundancy information is shown in figure 4.

A defined address is used in the packet header to identify that redundancy bytes are present. A SYNC byte

can also be used for this. In this context, no useful data can be transported in the packet.

In a cohesive data stream, the arrangement shown in figure 5 is obtained for this alternative.

In this example, the code is shortened to the length (208/192). The code word is not calculated using the packet header PH and the padding P.

10

In this embodiment, the cycle for the error protection useful data length must match that for the packets, i.e. the useful data length of the code word for the additional error protection must be a multiple of the minimum size for a packet length, in this case 24. The chosen useful data length is then filled with an integer combination of the four possible packet sizes. The packet with the redundancy information R is appended thereto. During decoding, only the redundancy bytes are used from the redundancy packet, and the rest is discarded completely (PHc, P, CRC).

A further exemplary embodiment based on the invention is shown in figure 6. In this case, the same REED-Solomon mother code RS (255, 239) is used as for DVB and DMB Video. If required, it is shortened by adding zeros before the protection bytes are calculated. The packets are arranged in rows. After 188 rows, for example, the code words are calculated in columns. This results in improved error correction performance, since burst errors within the packets are distributed over a plurality of code words.

The RS redundancy bytes (RS bytes in figure 6) are also accommodated in the DAB packet structure. To reduce the overhead, only a shortened packet header is used. The control bit is not necessary, and the full packet length is always used for the redundancy bytes. Figure 6 shows that the RS redundancy bytes in the first two

columns are placed at the end of the RS packets. Hence, the data capacity released in the shortened packet header is replaced by RS redundancy. bytes. Error protection over the CRC columns are not required.

5

A fully backward-compatible (level 2) solution for error correction for DAB packet mode with implied interleaving can be achieved in this way.

Patent Claims

1. A method for correcting errors in packet data, where code words are used over a plurality of data packets, having the following steps:
 - redundancy information for correcting errors is added while retaining the original packet data structure, specifically at the cost of a free data field or of a stuffing data field or useful data field,
 - the cycle of error protection, particularly its code word length, is chosen as a multiple of a minimum size for a packet length.
2. The method as claimed in claim 1, characterized in that the packet data used are DAB data packets.
3. The method as claimed in claim 1 or 2, characterized in that an outer code word length is provided as a multiple of 24 bytes or an inner code word length is provided as a multiple of 24 bytes.
4. The method as claimed in either of claims 2 or 3, characterized in that the error recognition information in the data packet containing the redundancy information for correcting errors is associated with the next respective code word.
5. The method as claimed in one of claims 1 to 4, characterized in that to recognize that redundancy information for correcting errors is contained in a data packet and/or to find the position of the code word in the data stream an appropriate identifier is provided in the packet header, e.g. a set command bit, or in the sync word.
6. The method as claimed in one of claims 1 to 5, characterized in that to correct errors a REED-Solomon code is used which is shortened particularly to the

packet lengths used.

7. The method as claimed in one of claims 1 to 6,
characterized in that a separate data packet is used
5 for the redundancy information, said data packet being
appended to the data packets for useful information.

8. The method as claimed in claim 7, characterized in
that only the redundancy information is evaluated for
10 decoding the data packet with the redundancy
information, and the rest is discarded.

9. The method as claimed in one of claims 1 to 8,
characterized in that the data packets are arranged in
15 rows and the code words in columns.

10. The method as claimed in claim 9, characterized in
that burst errors within packets are distributed over a
plurality of code words.

20

11. The method as claimed in either of claims 9 and
10, characterized in that a shortened packet header is
used, and the data capacity resulting from shortening
is filled with redundancy bytes.