Implementation of Error Resilient Video Transmission over Wireless Networks

Abhilash Reddy B*1, S.P.V.S Rao*2
# M.Tech & DSCE & Department of ECE & JNTU-Hyderabad
Plot No 17, Archna Pride flat#D4, Gachibowli,Hyderabad

Abstract-- Generally, digital compressed video is considerably vulnerable to transmission errors, due to the use of predictive and variable-length (entropy) coding. Even a single bit error can lead to damage of a whole block or set of video blocks. In contrast with simple data communications, which are not subject to strict delay conditions, real-time video transmission is delay sensitive and cannot make use of retransmission in order to ensure error-free delivery of the video data. So it requires careful design of the systems to recover the transmitted errors.

This paper deals with the implementation of video transmission using Matlab Simulink Model. We transmitted the video over Additive White Gaussian Noise (AWGN) channel by using the DVB-T transmitter standard and performed different channel coding techniques and analyzed the performance of video by observing the parameters BER, PSNR and SNR.

Keywords--- Compression, Channel Coding techniques, Video Transmission.

I. INTRODUCTION

For effective video communications, reduction of raw video rates, i.e. video coding, is only the first of the necessary steps. The compressed bit stream is segmented into fixed or variable length packets. The packets could be sent directly over the channel, if the channel guarantied error-free transmission. However, this is not the case. Therefore, the packets usually undergo a Channel Coding stage, using Forward Error Correction (FEC) techniques and Interleaving, in order to become more robust against transmission errors. At the receiver, the received packets are Channel Decoded and unpacked, before the resulting bit stream is decompressed.

Transmission errors can be roughly categorized into a) random bit errors and b) erasure errors. Erasure errors are mainly caused in packet networks, such as the Internet. In video communication systems, such as the Digital Video Broadcasting - Terrestrial (DVB-T) system, we will concentrate mainly in random bit errors. The use of Variable Length Coding (VLC) (e.g. Huffman) makes compressed video vulnerable to random bit errors, because a bit error can desynchronize the coded information and many following bits are wrongly decoded or become non-decodable. Due to the above reason and the fact that video compression techniques use predictive coding (e.g. a frame is predicted from a previous one in MPEG), error control in video communications is a very challenging task. Because of VLC, a single error can cause damage over a large portion of a video frame. Furthermore, due to the use of temporal prediction, a single erroneously recovered sample can lead to errors in future frames, i.e. the error propagates temporally.

II. CHANNEL CODING

Forward Error Correction (FEC)

In order to render the bit stream more resilient to bit errors, the transmitter must add redundancy (i.e. some additional bits) to the stream, so that error detection and correction to become possible. This is done using Forward Error Correction (FEC) techniques. However, notice that the addition of bit redundancy is opposite to the idea of video compression/coding which aims to the reduction of the bit rate. Therefore, generally, FEC must be carefully designed. Due to the same reason, the problem of joint video and channel coding is studied by the scientific community in the last years.

There are two main categories of FEC codes:

A. Block codes

This type of FEC codes works on fixed-length streams of bits. Practical block codes can generally be decoded in polynomial time to their stream length.

B. Convolutional codes

Convolutional codes work on bit streams of arbitrary length. They are most often decoded with the Viterbi algorithm. Viterbi decoding allows asymptotically optimal
decoding efficiency with increasing constraint length of the convolutional code, but at the expense of exponentially increasing complexity. Classical block codes and convolutional codes can be combined in concatenated coding schemes. In this case, a short constraint-length Viterbi-decoded convolutional code does most of the work. The block code (e.g. a Reed-Solomon code), with a larger block length, detect and correct any errors made by the convolutional decoder.

C. Interleaving

Interleaving is a way to rearrange digital data in a non-contiguous way and it is frequently used in digital communication systems in order to improve the performance of Forward Error Correcting codes.

In many cases of communication channels, errors typically occur in bursts rather than independently. Therefore, if the number of errors within a code word exceeds the error-correcting code's capability, the latter fails to recover the original code word. Interleaving helps us to override this problem by shuffling the source bits across several code words, thereby creating a more uniform distribution of errors.

III. IMPLEMENTATION METHODOLOGY

The system is composed of different fundamental blocks and user defined blocks and the functionality of those blocks are explained below.

The source video is compressed and decompressed using motion compensation and DCT transform coding standard techniques, and used the DVB-T standard coding technique for channel coding. Fig.2 represents the operation of DVB-T (Digital Video Broadcasting – Terrestrial) Transmitter it is a European-based consortium standard for the broadcast transmission of digital terrestrial television.

![](image)

**Source coding and multiplexing:**

Source video and audio are compressed. Compressed video, audio, and data streams are multiplexed into MPEG program streams. One or more program streams are joined together into an MPEG transport stream, which is the basic digital stream transmitted and received by our TV sets.

**Packetizing:**

The MPEG transport stream is split into a sequence of 188-bytes-long packets. The byte sequence is decorrelated using an “energy dispersal” technique.

**External FEC encoder:**

A first level of FEC encoding is applied to the transmitted data. A non-binary block code, a Reed-Solomon RS (204, 188) code is used. This code allows the correction of up to a maximum of 8 wrong bytes for each 188-byte packet.

**External interleaver:**

An External interleaver is used to rearrange the data sequence. A convolutional interleaver is used. The rearrangement is such that the transmitted data becomes more resilient to long sequences of errors.

**Internal FEC encoder:**

A second level of FEC error correction is applied. A punctuated convolutional code is used. There are five valid coding rates for DVB-T: 1/2, 2/3, 3/4, 5/6, and 7/8.

**Internal interleaver:**

An internal block interleaving technique is adopted to rearrange the data sequence again. This inner interleaver aims to reduce the influence of burst errors.

**Digital modulation mapper:**

QPSK, 16-QAM, or 64-QAM is used to map the bit sequence into a (base-band) sequence of complex symbols.

**Frame adaption:**

The complex symbols are grouped in blocks of constant length. A sequence of 1512, 3024, or 6048 symbols constitutes a block.

**OFDM modulation:**

The sequence of blocks is modulated using the OFDM technique, DVB-T uses 2048, 4096, or 8192 OFDM subcarriers (2k, 4k, 8k mode, respectively).

IV. SIMULATION & RESULTS

The system is simulated using the Matlab Simulink model in this we used the design of DVB-T Channel coder design and developed modules for video encoder, channel coder and video decoder. The video is compressed using motion compensation and discrete cosine transforms (DCT)
techniques and Transmitted using OFDM on AWGN noisy channel. For simulation I considered a video of 640x480, 13 sec duration with frame rate of 30 frames/sec, the simulation results are shown in the Table (1) are BER results for different SNR ranges, Table (2) are PSNR values for different SNR ranges respective plots for SNR vs BER and SNR vs PSNR are shown in Fig.3 and Fig.4. The snapshots of the video at different SNR values for all techniques are shown in Fig.5.

**Bit Error Rate (BER)**

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>Without channel coding</th>
<th>Reed-Solomon Encoder</th>
<th>Convolutional interleaving</th>
<th>Convolution encoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>5.354</td>
<td>5.220</td>
<td>5.0523</td>
<td>4.95</td>
</tr>
<tr>
<td>16</td>
<td>4.21</td>
<td>4.122</td>
<td>4.005</td>
<td>0.965</td>
</tr>
<tr>
<td>17</td>
<td>3.120</td>
<td>3.053</td>
<td>3.9</td>
<td>0.1856</td>
</tr>
<tr>
<td>18</td>
<td>1.85</td>
<td>1.789</td>
<td>2.02</td>
<td>0.0095</td>
</tr>
<tr>
<td>19</td>
<td>1.225</td>
<td>1.234</td>
<td>1.21</td>
<td>0.0001</td>
</tr>
<tr>
<td>20</td>
<td>0.785</td>
<td>0.650</td>
<td>0.632</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0.432</td>
<td>0.0445</td>
<td>0.407</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0.215</td>
<td>0.0043</td>
<td>0.1632</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>0.045</td>
<td>0.00</td>
<td>0.0056</td>
<td>0</td>
</tr>
</tbody>
</table>

**PSNR Values For Different Techniques**

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>Without channel coding</th>
<th>Reed-Solomon Encoder</th>
<th>Convolutional interleaving</th>
<th>Convolution encoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>-47.85</td>
<td>-47</td>
<td>-46.01</td>
<td>-41.5</td>
</tr>
<tr>
<td>16</td>
<td>-44.65</td>
<td>-45.23</td>
<td>-45</td>
<td>-32.5</td>
</tr>
<tr>
<td>17</td>
<td>-42.50</td>
<td>-41.5</td>
<td>-42</td>
<td>27</td>
</tr>
<tr>
<td>18</td>
<td>-41.50</td>
<td>-41</td>
<td>-41.5</td>
<td>42</td>
</tr>
<tr>
<td>19</td>
<td>-40.50</td>
<td>-40.5</td>
<td>-40</td>
<td>Inf</td>
</tr>
<tr>
<td>20</td>
<td>-40.63</td>
<td>-38</td>
<td>-38</td>
<td>Inf</td>
</tr>
<tr>
<td>21</td>
<td>-37.5</td>
<td>17.29</td>
<td>-36</td>
<td>Inf</td>
</tr>
<tr>
<td>22</td>
<td>-34.83</td>
<td>35.6</td>
<td>-29</td>
<td>inf</td>
</tr>
<tr>
<td>23</td>
<td>22.57</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
</tr>
</tbody>
</table>
With Convolutional Interleaving  
\( e) \) At SNR = 21dB  \( f) \) At SNR = 22dB

With Reed-Solomon code  
\( g) \) At SNR = 21dB  \( h) \) At SNR = 22dB

With Punctuated Convolutional code  
\( i) \) At SNR = 16dB  \( j) \) At SNR = 17 dB

Fig. 5 Snapshots of Different techniques at different SNR values  
(From (a to j))

V. CONCLUSION  

The paper concludes that the transmitted errors recovery with Convolution Encoder gives better results as compared to other technique and by seeing at the results we can say that convolutional encoder provide better quality of video at 5 dB lesser than all other techniques.

REFERENCES


