Unequal Error Protection (UEP) Solution for Video over 3G Wireless Channel

Allan J. Lee
Florida Atlantic University
Department of Electrical Engineering
allanjalee@hotmail.com

Abstract

The transmission of real time video over noisy time-varying channel is a very challenging task due to the unpredictable nature of the channel. A successful approach has been the use of forward error correction (FEC) in an unequal loss protection scheme along with a scalable video source to adapt the source data to the varying channel conditions. In this paper a method is explored which seeks to match an MPEG-4 FGS video signal to a third generation (3G) wireless channel using a product code. The MPEG-4 base layer (BL) is sent intact using equal error protection while the enhancement layer (EL) is sent using unequal error protection. The BL depends on the system being provisioned with a minimum QoS bandwidth. The product code used is a cyclic redundancy check (CRC) concatenated with a rate-compatible punctured convolutional code (RCPC) across the rows of a block of packets (BOP) and a Reed-Solomon (RS) code across the columns. The RCPC is used to correct bit errors and the CRC gives an indication of packet loss while the RS code is used to correct any packet erasures. System simulation using NS-2, shows that the source data can be gracefully adapted the varying channel conditions.

1. Introduction

In this paper, a method of transmission of video sequences over third generation (3G) mobile networks is investigated. The approach is based on the one taken in [1], in which a scalable video bitstream is adapted to the wired network conditions through an unequal forward error correction (FEC) scheme. The aim was to mitigate the effects of packet loss on the video quality at the decoder. The difference here is that modifications are adapted to deal with the high bit error characteristics of the wireless channel. Also, it is assumed that there is a minimum throughput that is guaranteed across the 3G channel.

The main difference between the transmission of video over wireless networks and its transmission over wired networks is the high level of noise that is encountered in the wireless networks. Packet loss is a limiting factor in both types of networks and it occurs in both networks but emerges from different causes. In wired networks the packet loss phenomena is primarily due to congestion at intermediate nodes, while in wireless networks packets are mainly lost due to heavy bit and burst errors. TCP has been widely applied in wired networks to combat the packet loss problem. Modified versions of the TCP protocol was applied in [2], [3], [4], to handle packet loss over wireless networks, however, the use of TCP incurs delays through these modifications, increasing processing time at the encoder. In addition to the packet loss problem, real-time video streaming is very challenging due to the delay constraints.

The approach in this paper is to use an embedded video bit stream to achieve video quality-scalability over the unstable and unpredictable wireless channel. The use of MPEG-4 FGS [5] serves our purpose exactly as it provides a base layer (BL) that is not scalable and an enhancement layer (EL) that is progressive and can be adapted to the wireless channel. The BL is non-scalable and must be received and decoded perfectly at the decoder, while the EL is embedded and can be truncated anywhere to achieve a desired bit rate. An unequal loss protection (ULP) scheme is used to protect the embedded EL against packet loss.

To provide unequal loss protection against packet loss, multiple description coding (MDC) systems [6] are largely used. The ULP scheme mentioned above is derived from this multiple description concept and can be used with an embedded bit stream to provide graceful degradation when the channel error rate is high. In the embedded bit stream, the earlier bits are more important (add more to the end quality) than the later bits. Considering this, MD systems provide ULP to the enhancement layer using FEC with multiple descriptions (different FEC) to protect different areas of the bit stream differently. Recovering any number of EL bytes will add to the BL quality, assuming that the BL is received and decoded perfectly.

Based on the requirements for IMT-2000 [7], data throughput is envisioned to be as high as 2Mbps. The assumption in this paper is that a throughput of 144Kbps will be guaranteed and represents a lower bound bandwidth across the 3G air-interface. The BL will then be protected with an equal loss protection scheme devised to maximize protection based on this throughput. If this assumption does not hold the BL quality will be substantially be degraded and the by the same token the EL will not be available, and the mobile client will experience outage. Guaranteed QoS scheme [8] can be used to ensure that the BL is received intact and the success of the proposed system is largely based on this guaranteed QoS system. One of the advantages of this system over the others mentioned earlier is that the variance of the decoder video quality will be limited by the lower bound on the throughput requirement. This will definitely improve the mobile viewer experience.

A similar scheme was proposed in [9], but the BL was protected with FEC as well as a retransmission scheme that ensures that the BL is received intact before streaming the EL information. This approach can cause decoding problems in real-time interactive video sessions.

In this paper, the following section takes a closer look at the bit stream definition of MPEG-4 FGS bit stream, and then in section 3, the wireless channel characteristics are outlined. Section 4 then discusses the channel coding based on the product code concept and in section 5 the byte allocation algorithm is outlined. In section 6 system simulation is carried out and the results discussed

.

2. Fine Granularity Scalability (FGS)

Fine granularity scalability has been introduced in the MPEG-4 standard specifically for the transmission of video over the internet [10]. Networked video communications presents some new challenges in the encoding of compressed video, in which the original objective was to optimize video quality at a given bit rate. This objective has now been redefined due to network video applications such as Internet video streaming and more recently, streaming to wireless clients. The channel capacity in both scenarios varies over a wide range depending on the type of connection and the network traffic at any given time in the case of the wired Internet, and in the wireless case due to channel fading.

In the traditional video communication system, the encoder compresses the input video signal into a bit rate that is less than the channel capacity and the decoder reconstructs the video signal using all the bits received from the channel. In this system the decoder process the bit stream as a constant-bit-rate signal and renders the reconstructed video. In the Internet and the wireless channel scenarios, the varying channel capacity would have starved the decoder of decoding bits when the channel capacity drops below the system target bit rate. To continue to render useful images in this scenario the decoder must be capable of decoding video over a range of bit rate. The bit stream therefore, should be partially decodable at any bit rate within a defined bit rate range to reconstruct a video signal with the optimized quality at that bit rate.

Several scalability techniques have been introduced to meet this bit stream requirement [11], they are referred to as layered coding. In these techniques several bit streams representing a single video signal is produced at the encoder, usually a standalone low quality base layer (BL) and several dependent enhancement layers (ELs). At the decoder, the base layer alone can be decoded in a low bit rate channel, rendering a low quality signal and with incremental increase in the channel capacity other higher layer can be received, in a hierarchical manner, and added to the base layer to render a progressively higher quality signal depending on how many enhancement layers are received and decoded. With these schemes each EL must be received and processed completely before it can have any meaningful effect on the decoded video. As a result, these schemes produce incremental improvements in video quality in coarse steps or with fairly large increase in the signal bit rate.

The unique characteristics of Fine Granularity Scalability (FGS) compared with the conventional layered coding is that the enhancement layer bit stream does not have to be received totally in order to increase the quality of the decoded signal. That is, the EL can be truncated anywhere and the remaining part can still add to the total video quality at the decoder.

2.1 MPEG-4 FGS

With MPEG-4 FGS the video is encoded into a base layer and one enhancement layer. Similar to the conventional layer coding, the base layer must be received completely in order to decode and display a basic quality video.

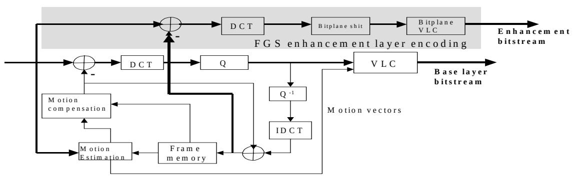

Figure 1: A Possible MPEG-4 FGS encoder

However, in contrast to the conventional scalable coding which requires the reception of the complete EL to improve upon the basic video quality, with FGS coding the enhancement layer can be cut anywhere before transmission. The received part of the EL stream cam be successfully decoded and improve upon the basic video quality. Similar to conventional layered coding, the FGS EL is hierarchical in that the higher order bits require the lower bits for successful decoding. When cutting the EL bit stream before transmission, the lower part of the bit stream (below the cut) needs to be transmitted and the upper part (above the cut) can be dropped. In FGS, the EL can be cut at the granularity of bits. Figure 1 above shows a possible encoder structure for MPEG-4 FGS. It is seen that the enhancement layer is derived by subtracting the BL from the original bit stream and applying bit plane coding afterwards, to generate the EL stream. The bit-plane coding is what gives the EL bit stream its embedded or progressive characteristics, which leads to the fine granularity property. The fine granularity characteristics enable FGS to be able to flexibly adapt to changes in the available bandwidth in wired and wireless networks. This property can also be exploited by video servers to adapt the streamed video to the available bandwidth in real-time (without required the computationally demanding re-encoding). In addition, the fine granularity property can be exploited by intermediate network nodes, such as base stations in wireless networks, to adapt video streams to the currently available downstream bandwidth. This is the approach that will be taken in this work with the use of MPEG-4 FGS. The MPEG-4 FGS signal will be protected on two distinct levels as it goes over the wireless network. The BL bit stream will be protected with an equal forward error protection (EFEP) scheme while the EL bit stream with be protected according to the packet loss characteristics of the channel, using an unequal forward error protection (UFEP).

3. 3G Wireless Channel

The proposal under IMT-2000 [7] for wireless communications specifies a theoretical maximum throughput across the wireless link of 2Mbit/s for the third generation mobile networks. This throughput value can only be achieved under ideal conditions. The wireless link suffers from high bit error rates and burst error due to the various forms of multipath fading, Doppler effects, climatic conditions, speed of the mobile client, the distance of the client from the serving base station, multipath fading, the number of users on the system (this occurs mainly with CDMA2000) and many other design or controllable factors. The end result is a highly error prone link which result in sever packet loss at times.

In this paper the basic 3G wireless channel model will be modeled as a memoryless packet erasure channel with throughput varying from 144Kbps to 2 Mbps.

4. Channel Coding

In our approach to adapt the video bit stream to the varying and unpredictable characteristics of the wireless channel, we will use a product code approach. A systematic Reed-Solomon (RS) code of maximal distance [12] will protect the transmitted symbols against erasure, while a rate-compatible punctured convolutional code (RCPC) concatenated with an outer CRC code [13], will protect the bits in each packet against bit errors. The product code will be applied across a block of packets (BOP), with the rows representing individual packets and individual blocks representing each byte within the packet. The columns will form RS-protected streams, containing both source data and FEC redundant symbols. The table 1 below depicts this setup.

Table 1: BOP with product code

1	2	3	6	9	13	17	+	+	0	0	0
X	X	4	7	10	14	18	+	+	0	0	0
X	X	5	8	11	15	19	+	+	0	0	0
X	X	X	X	12	16	20	+	+	0	0	0
X	X	X	X	X	X	21	+	+	0	0	0

In table 1 the rows represent the packets to be transmitted. The individual blocks represent the bytes within each packet. The blocks with numeric contents numbering 1 to 21 represent bytes of the source data, while the Xs represent redundant data generated by Reed-Solomon encoding. Each row (packet) is protected against bit errors by RCPC coding. The RCPC code is represented by the 0s in the table and finally, the integrity of the RCPC protection is checked by an outer CRC code, represented by the +.

The row codes can be decoded using the list-Viterbi algorithm [14], which selects the trellis path with the best match or metric that is used. If the wrong path is selected (this can happen due to bit errors), the result will be a packet with a lot of bit errors. The decoded bits are then subject to the constraints of the CRC check. The CRC code provides an indication of RCPC decoding success or failure of a packet. Decoding failures in the row codes are treated as erasures when decoding the column RS codes. The row codes will not be considered further in this paper since packet loss conditions of the channel will be treated as being the results of the CRC parity checker. If the CRC parity checker detects no errors, the packet is considered to be intact, otherwise, it is considered to be lost.

The systematic Reed-Solomon codes are used to implement the forward error correction (FEC) mechanism. The RS codes are effective against erased symbols when the locations of the erased symbols are known and the number of erased symbols does not exceed the number of redundant RS symbols. The RS codes will be optimized for erasures when packets arrive completely intact or are completely discarded. The RS code is a block code with maximum separable distance denoted by a pair (*N, k)*, where *N* is the block length (total number of symbols in the code) and *k* is the total number of source symbols. The RS code being systematic means that the first *k* symbols of the code are source symbols with the remaining *N*-*k* being RS redundant symbols. The code has the property that it can recover the *k* source symbols from any size-*k* subset of *N* total symbols for an (*N, k)* code. The positions of the erased symbols must also be known. The recovery is possible with this code by treating the source symbols as the coefficients of a polynomial in a Galois field of size $2^8 = 256$ and evaluating it at a number of additional points, thus creating redundant data [14]. Table 2 below depicts the scenario with the second packet being erased (errors detected by the CRC checksum).

Table 2: BOP showing packet loss

1	2	3	6	9	13	17	+	+	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?
X	X	5	8	11	15	19	+	+	0	0	0
X	X	X	X	12	16	20	+	+	0	0	0
X	X	X	X	X	X	21	+	+	0	0	0

This scenario arises when the outer CRC code detects an error in the packet number 2 after the RCPC decoding. With packet number 2 erased, the RS-code will be used to correct the streams affected. As stated earlier, a stream with an (*N*, *k*) RS code can recover from *N*-k or less symbol erasures. Looking at the scenario above it is readily seen that one symbol was erased from each stream as a result of the loss of packet number 2. The position of the erased symbol is also known (since the erased packet can be identified). For each stream *N =5* and *k* varies depending on the amount of protection offered for that stream. Applying this to the scenario above, we see that the lost symbols from streams one to six can be reconstructed since they have adequate RS protection. Specifically, bytes 4, 7, 10 and 14 can all be recovered. Only byte 18 was lost with the protection that was applied. The scenario with any number of packet erasures can be analyzed in a similar way.

4.1. Unequal Loss Protection

FEC is applied using the RS code in an unequal loss protection scheme. Since the bit-plane coded enhancement layer of the MPEG 4 video bit stream is progressive, different bytes in this bit stream have different effect on the resulting quality of the received signal. Specifically, the MSB bit plane is usually transmitted first, followed by the MSB-1, then the MSB-2 and so on. As a consequence, the earlier bytes within the bit stream (which are the MSB bytes) are more valuable in terms of their contribution to the decoded picture quality at the receiver. In our system, the channel coding is applied with this in mind. Based on work done in [15], streams (defined earlier) of the video data can be protected with higher priority than other streams. Since the bit stream is progressive, it can be truncated anywhere to facilitate the variable RS coding assignments, with the earlier bits still contributing to the final picture quality at the decoder.

The UEP framework will be based on the BOP elaborated on earlier. Message streams will be formed from a fixed number of bytes (the columns of the BOP). The algorithm described later will determine the amount of source data and FEC bytes that form each stream. Rate adaptation can be obtained by using feedback data on the channel characteristics as a measure by which to adjust the level of protection for each message stream that is generated. This scheme is depicted below in table 3.

Table 3

1	2	3	6	9	13	17
X	X	4	7	10	14	18
X	X	5	8	11	15	19
X	X	X	X	12	16	20
X	X	X	X	X	X	21

The above table elaborates the idea of the use of UEP to gracefully adapt the source data to the channel throughput. The block of packets will be of constant size for small variations of channel throughput, determined from the algorithm in [15] to match the average loss condition of the channel. Source data will be heavily coded (low source data rate) or lightly coded (high source data rate) depending on the channel throughput. This will result in a variable source data rate that will be protected according to the channel conditions. Adaptation to large bandwidth variations can be achieved through adjustment of the block of packet (BOP) size. The optimal BOP size can be determined by algorithm in [14].

5. Byte Allocation Algorithm

To formalize the ideas elaborated above, the algorithm used in [15] will be analyzed to determine the expected distortion at the decoder when both BL and the EL are received and decoded.

Table 4

Packet number							
1	1	2	3	6	9	13	17
2	X	X	4	7	10	14	18
3	X	X	5	8	11	15	19
4	X	X	X	X	12	16	20
5	X	X	X	X	X	X	21
Stream number	1	2	3	4	5	6	7

With reference to table 4, assume there is a sequence of data bytes of a message M (the numbered blocks) to be transmitted. This could be the corresponding enhancement layer of a coded video frame. Since we have a varying channel throughput available to the EL, at any point in time, instead of sending the total message M, a prefix of M is sent. The prefix of M and some FEC (Xs in the table) will make up the total bytes in the block. It is readily seen that the total byte rate is always the same but the source data rate changes as the amount of FEC changes. To introduce some notations, let m_i equal the number of data bytes assigned to stream i and let $f_i = N - m_i$ equal the number of FEC bytes for stream i. The redundancy assignment can then be represented as an L – dimensional FEC vector with entries that are the length (number of FEC bytes) of the FEC assigned to stream $i = 1 \text{ to } L$, as

$$\boldsymbol{f} = (f_1, f_2, \ldots, f_L).$$

For a given $\boldsymbol{f}$, we divide M into message fragments $M_i(\boldsymbol{f})$ and refer to $M_i(\boldsymbol{f})$ as the sequence of source data in stream i. $M_1(\boldsymbol{f})$ will then be the sequence of data bytes in stream 1. Now the prefix of M containing the first j fragments for redundancy vector $\boldsymbol{f}$ can be denoted as

$$M(j, \boldsymbol{f}) = M_1(\boldsymbol{f})M_2(\boldsymbol{f}) \ldots . M_j(\boldsymbol{f}).$$

With the notation above we can define the incremental PSNR of stream i as

$$g_i(\boldsymbol{f}) = PSNR[M(i, \boldsymbol{f})] - PSNR[M(i\text{-}1, \boldsymbol{f})]$$

This quantity $g_i(\boldsymbol{f})$ is the amount by which the PSNR increases when the receiver decodes fragment i, given that all the previous fragment have been already decoded.

Since the data is progressive, it is required that $f_i \geq f_{i+1}$, resulting in an FEC assignment that is non-increasing with i. This requirement ensures that if $M_i(\boldsymbol{f})$ can be decoded, then $M_1(\boldsymbol{f})$, $M_2(\boldsymbol{f})$ up to $M_{i-1}(\boldsymbol{f})$ must already have been decoded.

To determine the FEC code vector $\boldsymbol{f}$, an estimate of the channel loss profile that a message is likely to encounter is taken into consideration. For real time video communications this can be a metric that is monitored in real time and the information made available at the encoder at the time of evaluating the vector $\boldsymbol{f}$. Let p_n be the probability that n packets are lost, where $n = 0, 1, 2, \ldots, N$ and N is the total number of packets in a stream. The cumulative distribution function $c(k) = \Sigma^{k}_{n=0}\, p_n$; $k = 0, 1, N$, represents the probability that k or fewer packets are lost. With this information the probability that the decoder can decode stream *i* can be determined and is represented by the quantity $c(f_i)$. The expected PSNR of the received message as a function of $\boldsymbol{f}$ can be calculated by;

$$G(\boldsymbol{f}) = \Sigma^{L}_{i=1} c(f_i)\, g_i(\boldsymbol{f}).$$

In designing the algorithm to assign FEC, the $\boldsymbol{f}$ that maximizes $G(\boldsymbol{f})$ subject to a given packet loss p_n, is usually selected as the optimal solution.

6. Simulation

An attempt was made to simulate the system *NS-2* [16]. I had problems getting the NS-2 tcl code to work and then finally I ran out of time. The description of the simulation setup and the expected results will be discussed in this section.

Both the base layer the enhancement layer bit streams were would be simulated by two constant bit rate sources. Aim of the simulation was basically to test the concept of sending a derived redundant code with the source data to combat the effects of different channel packet loss condition. A node was use to combine the traffic from both sources, this represents the base station streaming mechanism. Traffic from both sources will be combined and sent to the receiving node, this represents the mobile terminal receiver processor. This setup is outlined in figure 2 below.

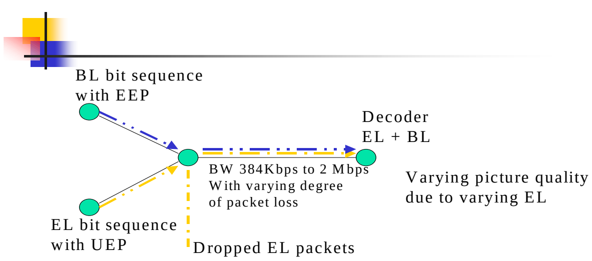

Figure 2: NS-2 simulation setup

The idea is to replicate the effect of sending systematic Reed-Solomon codes along with the source data. Both the BL and EL bit stream generators will send a sequence of bytes at a constant rate. For both bit stream sequence, the total bytes sent will be allocated between source data and FEC redundant symbols. For the BL stream, a fix assignment will be used, while for the EL sequence, the distribution of source bytes to FEC bytes will be determined from the channel loss conditions. The BL sequence will be guaranteed a certain portion of the available bandwidth (replicating a guaranteed QoS for BL), while the EL sequence will be streamed according to the remaining portion of the bandwidth. Each flow will be monitored separately; the BL flow and the EL flow. The output of the simulation will be the number of packets received for each flow. When this output is matched up against the number of packets sent, then based on the FEC code assigned at the time, it can be determined if all the source data can be received.

Now for a particular MPEG-4 trace, the source-rate distortion curve (SRD) would give the expected PSNR at the decoder depending on the rate of the EL stream.

The difficulties I had was to configure the NS-2 code to prioritize the BL stream and give it a guaranteed throughput, to monitor and record statistics about each flow separately, such as the packet loss information and finally to implement the channel error model. Had time permitted, these could have been figured out and output data be made available to indicate the performance of the system.

7. References

[1] Uwe Horn, K. Stuhlmüller, M. Link and B. GirodA.B., “Robust Internet video transmission based on scalable coding and unequal error protection”, Signal Processing: Image Communication, Volume 15, Issues 1-2 , September 1999, Pages 77-94.

[2] M. Allman, D. Glover, L. Sanchez, “RFC 2488, Jan. 1999.

[3] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, R. KaKatz, “A comparison of Mechanisms for improvingTCP performance over Wireless Links”, ACM SIGCOMM, Aug. 1996.

[4] C. Barakat, E. Altman, W. Dabbous, “On TCP performance in a Heterogeneous Networks: A Survey”, IEEE Communications Magazine, vol. 38, no. 1, pp 40-46, Jan. 2000.

[5] Li, W. “Overview of fine granularity scalability in MPEG-4 video standard”, IEEE Trans. On Circuits and Systems for Video Technology, vol. 11, pp 301-317, March 2001.

[6] Puri, R., Ramchandran, K., “Multiple description coding using forward error correction codes”, Proc. 33rd Asilomar Conf. On Signals and Systems, vol. 1, pp. 342-346, Pacific Grove, CA. Oct. 1999.

[7] Raj Pandya, et al, "IMT-2000 Standards: Network Aspects", Personal Communications, IEEE, Volume: 4 , Issue: 4, pp. 20-29, Aug.1997.

[8] Sung-Hyun Cho, Seung-Jei Yang, Shin Heu, Sung-Han Park, "QoS oriented bandwidth management scheme for stable multimedia services on the IMT-2000 networks", Information Technology: Coding and Computing, 2000.

[9] Youssef Charfi, Raouf Hamzaoui, "Packet Loss Protection of Scalable Video Bitstreams Using Forward Error Correction and Feedback", www.inf.uni-konstanz.de/~charfi/ChHa_ISPA03.pdf.

[10] Philippe de Cuetos, Martin Reisslein, "Evaluating the streaming of FGS-encoded Video with Rate-Distortion Traces", Institut Eurecom Technical Report, June 2003. http://trace.eas.asu.edu/indexfgs.html.

[11] Yao Wang, et al, *Video Processing and Communications*, Prentice Hall, Upper Saddle River, New Jersey, 2002.

[12] L. Rizzo, "Effective erasure codes for reliable computer communications protocols", ACM Computer Communication Review, vol. 27, no. 2, pp 24-36, April 1997.

[13] D. G. Sachs, R. Anand, and K. Ramchandran, "Wireless image transmission using multiple-description based concatenated code," Proc. SPIE Image and Video Communications and Processing, vol. 3974 pp. 300-311, San Jose, CA, Jan. 2000.

[14] Vladimir Stankovi´c, Raouf Hamzaoui, "Live Video Streaming over packet networks and wireless channels", www.polytech.univ-nantes.fr/pv2003/ papers/pv/papers/cr1020.pdf.

[15] A. E. Mohr, E.A. Riskin, and R.E. Ladner," Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction," IEEE Journal on Selected Areas in Comm., vol. 18, no. 7, pp. 819-828, Dec. 2000.

[16] The LBNL Network Simulator, NS-2, http://www.isi.edu/nsnam/ns/.