Error Resilient Interactive Video Streaming
Over Wireless Networks

Marios G. Hadjinicolaou¹, Kostas E. Psannis² and Yutaka Ishibashi ³
¹ Department of Electronic and Computer Engineering,
School of Engineering and Design, Brunel University, UK
² Department of Technology Management
University of Macedonia, Thessaloniki 540 06, Greece
³ Department of Computer Science and Engineering,
Nagoya Institute of Technology, Nagoya 466-8555, Japan
Corresponding Author: ² kpsannis@uom.gr ³ mobility2net@gmail.com

ABSTRACT

This paper presents an efficient approach for supporting wireless video full interactive services. One of the main goals of wireless video multicast services is to provide priority including dedicated bandwidth, controlled jitter (required by some real-time and interactive traffic), and improved loss characteristics. The proposed method is based on storing multiple differently encoded versions of the normal/interactive video streams at the server. The corresponding video streams are obtained by encoding the original uncompressed video file as a sequence of I-P(I) frames and I-P(M) frames using different GOP (Group Of Pictures) pattern. Mechanisms for controlling the normal/interactive request are also presented and their effectiveness is assessed through extensive simulations. Wireless normal/interactive video services are supported with considerably reduced additional delay and acceptable visual quality at the wireless client–end.

Keywords—Wireless Interactive Applications, Streaming Wireless Video, Energy Management

1. INTRODUCTION

The demand for media rich applications over the Internet as well as wireless networks has rapidly increased in recent years. Video transport is becoming increasingly important for a large variety of applications and networks. The current mobile cellular systems, which are also known as second generation (2G) systems, provide a basic 9.6 Kbps data rate (e.g. GSM systems). These 2G systems have recently been upgraded to increase their capabilities. For example, General
Packet Radio Systems (GPRS) provide up to 115 Kbps data rates, and the Enhance GPRS (EGPRS) further support the data rates up to 384 Kbps when eight times slots are used. [1] The emerging third generation (3G) systems, also known as International Mobile Telecommunications-2000 (IMT-2000), provide up to 2Mbps data rates. Moreover, the next generation wireless systems (4G), which is currently in the design phase, will support even much higher data rates. Applications that are in the process of becoming essential to users include video telephony, gaming, or TV broadcasting. This trend creates great opportunities for identifying new wireless multimedia applications, and for developing advanced systems and algorithms to support these applications. Given the nature of the channel and of the mobile devices, issues such as scalability, error resiliency, and energy efficiency are of great importance in applications involving multimedia transmission over wireless networks. When developing a reliable and effective video transmission system over a wireless-based network, many different technical problems must be addressed, some of which may be application- or network-specific. With the rapid increase in the bandwidth of wireless networks and the popularity of broadband Internet access, media rich services such as interactive multimedia and video conferencing are expected to be widely used in the near future. Users will be able to receive almost the same services on their wireless systems and wireline networks. Enabling these media rich communications services requires not only enhanced broadband access, but also powerful media coding methods to make the transmission more efficient [2].

In recent years several error resilience techniques have been devised [3]-[9]. In [3], an error resilience entropy coding (EREC) have been proposed. In this method the incoming bitstream is re-ordered without adding redundancy such that longer VLC blocks fill up the spaces left by shorter blocks in a number of VLC blocks that form a fixed-length EREC frame. The drawback of this method is that the codes between two synchronization markers are dropped. This results any VLC code in the EREC frame be corrupted due to transmission errors. In [4] an error resilience transcoder for General Packet Radio Service (GPRS) mobile accesses networks is presented. In this approach the bit allocation between insertion error resilience and the video coding is not optimized. In [5] optimal error resilience insertion is divided into two subproblems: optimal mode selection for macroblocks and optimal resynchronization marker insertion. Moreover in [6] an approach to recursively compute the expected decoder distortion with pixel–level precision to account for spatial and temporal error propagation in a packet loss environment is proposed. In these methods [5] and [6] the inter frame dependencies are not considered. In MPEG-4 video standard, application layer error resilient tools were developed. At the source coder layer, these tools provide synchronization and error recovery functionalities. Efficient tools are
Resynchronization Marker and Adaptive Intra frame Refresh (AIR). The marker localizes transmission error by inserting code to mitigate errors. AIR prevents error propagation by frequently performing intra frame coding to motion domains. However AIR is not effective in combating error propagation when I-frames are less frequent.

A survey of error resilient techniques for multicast applications for IP-based networks is reported in [7]. It presents algorithms that combine ARQ, FEC and local recovery techniques where the retransmissions are conducted by multicast group members or intermediate nodes in the multicast tree. Moreover video resilience techniques using hierarchical algorithms are proposed where transmission of I-, P- and B frames are sent with varying levels of FEC protection. Some of the prior research work on error resilience for broadcast terminals focuses on increasing FEC based on the feedback statistics for the user [8]. A comparison of different error resilience algorithms for wireless video multicasting on wireless local area networks is reported in [9]. However in the literature survey none of the methods applied error resilience techniques at the video coding level to support wireless video services with full interactive functions. Developing error resilience technique which provides high quality of experience to the end mobile user is a challenge issue. Moreover several techniques for supporting interactivity for MPEG code video streaming applications have been devised [10]-[14]. A recent survey of interactive operations is reported in [15]. In this paper we propose a very efficient error resilience technique for streaming normal/interactive video streams over wireless networks. Encoding separate copies of the video, the normal/interactive video streams are supported with minimum additional resources and acceptable visual quality at the wireless client-end. The corresponding versions are obtained by encoding (i.e. uncompressed) frame of the original movie as a sequence of I-, P(I)- frames and I-P(Marionette)- frames respectively using different GOP (Group Of Pictures) pattern.

The paper is organized as follows. In Section 2 problems of supporting Normal/Interactive Wireless Video Services over wireless networks are discussed. In Section 3 the preprocessing steps required to support efficient normal/interactive video streaming services over wireless networks are detailed. Section 4 presents the extensive simulation results. In Section 5 we compare our algorithm for supporting full interactive functions against other schemes. Finally conclusions are discussed in Section 6.

2. PROBLEM DESCRIPTION

In the last decade video compression technologies have evolved in the series of MPEG-1, MPEG-2, MPEG-4 and H.264. Given a bandwidth of several hundred of kilobits per second, the recent codecs, such as MPEG-4, can efficiently
transmit high-quality video. An MPEG video stream comprises intra-frames (I), predicted frames (P), and interpolated frames (B). According to MPEG coding standards, I-frames are coded such that they are independent of any other frames in the sequence; P-frames are coded using motion estimation and each one has a dependency on the preceding I- or P-frame; finally the coding of B-frames depends on the two “anchor” frames - the preceding I/P frame and the following I/P frame. An MPEG coded video sequence is typically partitioned into small intervals called GOP (Group Of Pictures). Two parameters define the MPEG structure N and M. N is the distance between two successive I frames, defining a “Group of Pictures” (GOP). M is the distance between consecutive I or P frames.

2.1 Normal Video Streaming

When a mobile joins an existing normal session, there is a delay before which it can get synchronized. This delay is proportional to the frequency of I-frames as determined by the streaming server. Since I-frames require more bits than the P- and B-frames, the compression efficiency is inversely to the frequency of I-frames. Assume that \( I_{\text{number}} \) is the frequency of I-frames, \( F_{\text{rate}} \) is the frame rate of the video compression. The worse case initial delay in seconds can be computed as follows

\[
\text{Delay} = \frac{1}{I_{\text{number}}} - \left( \frac{1}{F_{\text{rate}}} \right)
\]

where \( I_{\text{number}} = \frac{F_{\text{rate}}}{N} \) and \( N \) is the distance between two successive I frames defining a “Group of Pictures” (GOP).

Fig.1 depicts the worse delay in seconds for different combination of frame rate and the number of I-frames in a Group Of Pictures.

The application would also require more frequent transmission of I-frames so as to allow the user to join the ongoing session. However this would result in requiring more bandwidth. The MPEG bitstream used for simulation is the “Mobile” sequence with 180 frames. Assume that \( M=3 \) (M is the distance between two successive P frames) for I-P-B structure. The average bandwidth is given by

\[
\text{Bandwidth} = F_{\text{rate}} \times \text{Average (IPB) Size} \times \frac{8 \text{bits}}{\text{byte}}
\]

where

\[
\text{Average (IPB) Size} = \frac{I_{\text{average}}}{N} + P_{\text{average}} \times \left( \frac{1}{M} \right) + B_{\text{average}} \times \left( \frac{1}{M} \right)
\]
averageI = Average size of Intra-frames,  
averageP = Average size of Predicted-frames and  
averageB = Average size of Bidirectional frames.

Eq. 1

Fig. 2 shows the increase in the network bandwidth as a function of Group Of Pictures. It can be seen from this graph that, more I- frames in a GOP results in increase in the network bandwidth. One other tool that is effective against error propagation is intra block refresh technique [5]. In this technique, a percentage of P- and B-frames block are intra coded and criterion for determining such intra clock is dependent on the algorithm. However the intra blocks refresh technique is not effective in combating error propagation when I- frames are less frequent.

2.2 Interactive Video Streaming

Implementation of the full interactive functions with the MPEG coded video presents a number of considerable challenges associated with video data storage and playout. An MPEG video stream comprises intra-frames (I), predicted frames (P), and interpolated frames (B). An MPEG coded video sequence is typically partitioned into small intervals called GOP (Group Of Pictures). This structure allows a simple realization of forward (normal)-play operation but imposes several additional constraints on the other interactive functions. Implementation of the rewind operations requires that the decoder either decodes the whole GOP and store it or it decodes the GOP up to the current frame to be displayed. Both of these techniques lead either to the requirement for large storage at the client machine (to store a fully decoded GOP) or massive decoding processing power (to fully decode a GOP at required displayed frame rate). Neither of these options is desirable. Furthermore, Fast Play (Fast Forward/Jump Forward) functions of MPEG coded video
present are problematic. When a P-/B- frame is requested, all the related previous P-/I-frames need to be sent over the network. This is likely to lead in considerable increase in communication bandwidth and decoding power compared to the normal-play mode. Assume that $I_i$ is the starting point of the fast playback mode and $F_s$ (Frames-skipped) is the number of skipped frames. Consider the case in which the current frame is $P_3$ and frame $P_9$ is the following frame to be displayed during the Fast Playback for $F_s$ factor equals to 6. Actually there is no need to send any of the B-frames because for the given $F_s$ factor the displayed P-frames can be decoded only from the preceding -I or -P frames. Fig.3 (c) depicts the number of required frames to be transmitted in order to display correctly the P-frames. It is valuable at this stage to develop a mathematical model in order to show the effect of the Fast Forward mode on the server load /network bandwidth and decoder memory. It is logical to assume that the start point of the Fast Forward mode is an I-frame. This is due to the fact that the I-frames will not cause an unpleasant effect in viewing because they are decoded independently.

$$I_o \ B_1 \ B_2 \ P_3 \ B_4 \ B_5 \ I_6 \ B_7 \ B_8 \ P_9$$

(a) normal play

$$B_1 \ B_2 \ B_4 \ B_5 \ B_7 \ B_8 \ B_{10} \ B_{11}...$$

(b) redundant frames

$$P_3 \ I_6 \ P_9 \ I_{12} \ P_{15} \ I_{18} \ P_{21} \ I_{24} \ P_{27}$$

(c) transmitted frames

Fig. 3. Fast Forward mode for $F_s$=6

A GOP is used as the retrieval block; normally stored contiguously on disk. Assuming that every I/O cycle, 1 GOP (15 frames) is retrieved, the disk executes one rotation, seek and transfer. If four different frames from three GOP (12 frames) and three frames form the next GOP are retrieved during each cycle, the disk must perform four rotations, seeks and retrievals for every 15 frames which increases the overhead (load) on the server to process Fast Forward (FF) request. Fig. 4 shows the increase on the load of the server, as a function of Frame-skipped ($F_s$) factor. The MPEG bitstream used for simulation is the “Mobile” sequence with 180 frames, which was encoded at 200Kbps, with a frame rate of 30fps. The Group Of Pictures format was $N=15$ and $M=3$.

Fig. 5 depicts the average bit rates required for sending the video stream with respect to different $F_s$ factors. The server needs to send extra frames to the decoder to display one frame. This results in increasing the network bandwidth. The
above analysis has been focused in the Fast-Forward operation. The results can be easily extended in the case of Fast-
Rewind (FR). This is due to the fact that the MPEG stream is symmetrical in the forward and backward directions.

3. PROPOSED METHOD

3.1 Normal Video Streaming
The problem addressed is that of transmitted a sequence of frames of stored video using the minimum amount of energy
subject to video quality and bandwidth constraints imposed by the wireless network. Assume that I- frame is always the
start point of a joining normal/interactive session. Since I- frames are decoded independently, switching from leaving to
joining normal/interactive session can been done very efficiently. The corresponding video streams are obtained by
encoding the original uncompressed video file as a sequence of I-P(I) frames using different GOP pattern
\(( N = 5, M = 1)\).

P(I) are coded using motion estimation and each one has a dependency only on the preceding I- frame. This result that
corruption of P-frame do not affect the next P-frame to be decoded. On the other hand, it increases the P(I) frame sizes.

We consider a system where source coding decisions are made using the minimum amount of energy

\[
\min E^{q(i)}\{I,P(I)\} \text{ subject to minimum Distortion } \left( D_{\text{min}} \right) \text{ at the mobile client and the available Channel Rate} \\
(C_{\text{Rate}}) \text{ required by wireless network. Hence}
\]

\[
\min E^{q(i)}\{I,P(I)\} \leq C_{\text{Rate}} \tag{1}
\]

\[
\min E^{q(i)}\{I,P(I)\} \leq D_{\text{min}}
\]
Our goal is to properly select a quantizer \( q(i) \) in order to minimize the energy required to transmit the sequence of I- P(M) frames subject to both distortion and channel constraints.

### 3.2 Interactive Video Streaming

The corresponding interactive video streams are obtained by encoding the original uncompressed video file as a sequence of I-P(M) frames using different GOP pattern \( (N_{\text{total}} = 5, M_{\text{total}} = 1) \). P(M) are coded using motion estimation and each one has a dependency only on the preceding I-frame. This result that corruption of P-frame do not affect the next P-frame to be decoded. On the other hand, it increases the P(M) frame sizes. We consider a system where source coding decisions are made using the minimum amount of energy

\[
E_{\text{min}} \{I, P(M)\} \leq D_{\text{min}}
\]

Our goal is to properly select a quantizer \( q(i) \) in order to minimize the energy required to transmit the sequence of I-P(M) frames subject to both distortion and channel constraints.

### 3.3 Control Scheme

A common approach to control the size of an MPEG frame is to vary the quantization factor on a per-frame basis. The quantized coefficients \( QF\{u,v\} \) are computed from the DCT coefficients \( F\{u,v\} \), the quantization_scale, \( MQUANT \), and a quantization_matrix, \( W\{u,v\} \), according to the following equation.

\[
QF \{u,v\} = \frac{16 \times F\{u,v\}}{MQUANT \times W\{u,v\}}
\]

The quantization step makes many of the values in the coefficient matrix zero, and it makes the rest smaller. The result is a significant reduction in the number of coded bits with no visually apparent difference between the decoded output and the original source data [22]. The quantization factor may be varied in two ways.

- Varying the quantization scale (\( MQUANT \))
- Varying the quantization matrix (\( W\{u,v\} \))

The encoding algorithm in the first encoding attempt starts with the nominal quantization value that was used to encode the preceding I-frame. After the first encoding attempt, if the resulting frame size fulfills the constraints (1)(2), the
encoder proceeds to the next frame. Otherwise, the quantization factor (quantization_matrix, $W[u,v]$) varies and the same frame is re-encoded. The quantization matrix can be modified by maintaining the same value at the near-dc coefficients but with different slope towards the higher frequency coefficients. This procedure is repeated until the size of the compressed frame corresponds to $(1)/(2)$. The advantage of this scheme is that it tries to minimizes the fluctuation in video quality while satisfying channel condition.

Fig. 7 shows two matrices both with the same value at the near-dc coefficients but with different slope towards the higher frequency coefficients. In other words, the quantization_scale is fixed $MQUANT$ and the quantization_matrix $W[u,v]$ varies.

4. EXPERIMENTS

There are two types of criteria that can be used for the evaluation of video quality; subjective and objective. It is difficult to do subjective rating because it is not mathematically repeatable. For this reason we measure the visual quality of the interactive mode using the Peak Signal-to-Noise Ratio (PSNR). We use the PSNR of the Y-component of a decoded frame. The PSNR is obtained by comparing the original raw frame with its decoded version with encoding being done using the proposed algorithms.

The MPEG-4 bitstream used for simulation is the “Mobile” sequence with 180 frame and frame rate of 30fps.

4.1. Normal Video Streaming

The GOP format was $N=15$, $M=1$ with the I-P(I) structure. We consider a set of allowable channel rate, $C_{Rate}=(300Kbps, 200Kbps, 100Kbps)$. In order to illustrate the advantage of the proposed algorithm we consider a system where source coding decisions are made without any constraints, using the same GOP format ($N=15$, $M=1$). Figure 8 shows the PSNR plot per frame obtained with the proposed algorithm and the reference scheme for the allowable
channel rate. Clearly the proposed algorithm yields an advantage of (PSNR) 1.42dB, 1.39dB, and 1.35dB, for the allowable channel rates 300kbps, 200kbps, 100kbps respectively.

4.2 Interactive Wireless Video Streaming

The GOP format was $N_{int} = 5, M_{int} = 1$ with the I-P(M) structure. We consider a set of allowable channel rate, $C_{Rate} = (300$Kbps, 200Kbps, 100Kbps). In order to illustrate the advantage of the proposed algorithm we consider a system where source coding decisions are made without any constraints, using the GOP format $N_{int} = 5, M_{int} = 1$. Let $T_F$ and $T_{F IP(M)}$ be the number of frames in the normal and interactive version respectively. The total frames in the interactive mode can be computed as follows

$$I = \frac{T_F}{N} = 12$$, 1- frames in normal mode. Hence $T_{F IP(M)} = I \times N_{int} = 60$ frames. Figure 9 shows the PSNR plot per frame obtained with the proposed algorithm and the reference scheme for the allowable channel rates 300kbps, 200kbps, 100kbps respectively.

5. COMPARATIVE STUDY

In recent years several techniques for supporting interactivity for MPEG code video streaming applications have been devised [16]-[25]. In [16], [17] interactive functions are supported by dropping parts of the original MPEG-2 video stream. Alternatively interactive functions can also be supported using separate copies of the movie that are encoded at lower quality of the normal playback copy [18], [19]. In these cases, there is no significant degradation in the visual quality. However, the number of pre-stored copies of the movie limits the speed-up granularity. Other conventional schemes that support interactive functions require that frames are displayed at a rate much higher than the normal playback [20], [21]. In the latter case the downloading can be done prior to viewing. Other approaches [22], [23], address only the issue of reverse functions of MPEG-2 video streams; specifically, they describe methods of transforming an MPEG-2 I-B-P compressed bitstream into I-B and I-P(M) bitstreams at client respectively. Some researchers suggested a model in which the server can partition each video stream into two sub-streams [24]. During the interactive mode, only the low-resolution stream is transmitted to the client. Moreover, [25] proposed a dual bitstream least cost scheme to efficiently provide VCR like functionality for MPEG-2 video streaming. Note that none of the methods mentioned above fully address the problem of supporting interactive operations with minimum additional resources at the load of the
server/network bandwidth and decoder complexity. In addition they support limited interactive functions. We compare our approach to the following approaches (a) Send only I-P- frames [17], (b) Partial GOP skipped [18], (c) Partition I-P- frames [24], (d) Dual Bit Stream at the server[25], (e) \((P\rightarrow I)\)Conversion [22]. The other approaches required additional decoding power which is not feasible using the decoders available today or in the foreseeable future. The comparison is performed with respect to the factors in the first row of table 1.

- Client resources refer to the memory and CPU requirements that are needed to process and decode a received frame.
- Interactive functions refer to the range of the supported operations
- Network bandwidth refers to the amount of bandwidth that is needed to support interactive functions.
- Additional Storage refers to the extra storage required to support interactivity

Because of the difficulty to quantify certain factors and the lack of detailed information about other approaches we contend with a qualitative comparison. The comparison is only meant to convey the tradeoffs provided by different schemes. Table 1 depicts that only the proposed very efficient method supports full range of interactive functions with minimum additional resources at the server load, network bandwidth and decoder complexity. In addition all the other approaches support limited interactive functions.

6. CONCLUSIONS

Error resilient (re)-encoding is a key technique that enables robust streaming of stored video content over noisy channels. It is particularly useful when content has been produced independent of the transmission network conditions. In this paper, we investigated the constraints of supporting normal /interactive multimedia services in wireless clients. In order to overcome these additional resources we proposed the use of multiple differently encoded version of each video sequence. The differently coded video is obtained by encoding frames of the original (uncompressed) sequence as a I-P(I) frames using different GOP pattern for the normal wireless stream. On the other hand, the interactive wireless streams are obtained by encoding every Nth frame of the original (uncompressed)- file as a sequence of I-P(Marionette)-frames. This allows for support of fully interactive operations at variable speedups. All versions of the original sequence are stored in the system’s server. The server responds to an interactive request by switching from the currently transmitted version to another version. By proper encoding versions of the original video sequence, video streaming services can be supported with considerably reduced additional delay and minimum jitter which implies acceptable visual quality at the wireless client –end. Moreover full interactive wireless video streams can be supported with considerably reduced additional storage, network bandwidth and decoder complexity respectively.

References


<table>
<thead>
<tr>
<th>Proposed Method</th>
<th>Client Resources</th>
<th>Interactive Functions</th>
<th>Network Bandwidth Allocation</th>
<th>Additional Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No increase at the client resources</td>
<td>Full_range (Backward Fast Jump/Forward/Rewind Slow Down Slow Reverse)</td>
<td>No extra bandwidth</td>
<td>Small additional storage (10% increase)</td>
</tr>
<tr>
<td>Send only P-frames</td>
<td>No increase at the client resources</td>
<td>Fast Forward</td>
<td>45% increased</td>
<td>No extra Storage</td>
</tr>
<tr>
<td>Partial - GoP-Skipped</td>
<td>Entire GoP buffered at the decoder</td>
<td>Fast Forward Jump Forward</td>
<td>41% increased</td>
<td>No extra storage</td>
</tr>
<tr>
<td>Skipped frames</td>
<td>Decoder stores frames</td>
<td>Fast Forward Jump Forward Fast Rewind Jump Rewind</td>
<td>No extra bandwidth</td>
<td>48% Increased</td>
</tr>
<tr>
<td>Dual bit stream at the server</td>
<td>Decoded frames/ increases decoder resources</td>
<td>Rewind, Fast Forward Fast Rewind</td>
<td>41% increased</td>
<td>100% increased</td>
</tr>
<tr>
<td>(P → I) Conversion</td>
<td>Buffers one frame</td>
<td>Rewind Fast Rewind</td>
<td>No extra bandwidth</td>
<td>28% increased</td>
</tr>
</tbody>
</table>

Table 1: Comparison of different approaches