World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:1, No:8, 2007

Bridging the Gap Between CBR and VBR for H264 Standard

Othon Kamariotis

Open Science Index, Electronics and Communication Engineering Vol:1, No:8, 2007 publications.waset.org/8876.pdf

***Abstract*—**This paper provides a flexible way of controlling Variable-Bit-Rate (VBR) of compressed digital video, applicable to the new H264 video compression standard. The entire video sequence is assessed in advance and the quantisation level is then set such that bit rate (and thus the frame rate) remains within predetermined limits compatible with the bandwidth of the transmission system and the capabilities of the remote end, while at the same time providing constant quality similar to VBR encoding. A process for avoiding buffer starvation by selectively eliminating frames from the encoded output at times when the frame rate is slow (large number of bits per frame) will be also described. Finally, the problem of buffer overflow will be solved by selectively eliminating frames from the received input to the decoder. The decoder detects the omission of the frames and resynchronizes the transmission by monitoring time stamps and repeating frames if necessary.

***Keywords*—**H264, CBR, VBR, Video Streaming, Digital Video, Multimedia, Buffering, Encoding, Decoding, Compression, Video-On-Demand

I. Introduction

IN the last few years, Broadband Internet Connections (DSL, Cable, Satellite) have become mainstream and more affordable to households. At the same time, there has been a growth in '3G Mobile Internet Services' and it is expected that they will become more popular, especially in the near future. These large 'pipes' of bandwidth have shown the way for more broadband services, like 'Video Streaming', 'HDTV' and 'Video-On-Demand', thus giving extra value to 'Broadband Internet' and exploiting the extra bandwidth provided to broadband customers. One of the key-technologies that helped in the expansion of the 'Internet' was digital video compression.

The need for better video compression and compatibility between digital video formats, resulted in the development and establishment of some international standards like 'H264' (latest), 'MPEG-4', 'MPEG-2', and 'H263'. One common feature of these standards is the fact that they only provide [2] the syntax of the compressed video stream; in other words, the output bit stream can be either constant-bit-rate (CBR) or variable-bit-rate (VBR).

Manuscript received September 15, 2005. This work was supported by BT Group.

Othon Kamariotis is with Broadband & Applications Research Center, British Telecom Group, BT Adastral Park, IP5 3RE, UK (phone: 0044-1473-643662; fax: 0044-1473-640929; e-mail: othon.kamariotis@ bt.com).

Since many digital video applications (especially 'Video Streaming') are constrained by constant limited channel bandwidth (like 'DSL' or 'dial-up' connections) or fixed storage size ('Personal Video Player', or 'DVD Recorder'), (CBR) encoding has been widely adopted because of its [2] practical implementation, ease of use and flexibility over 'IP Networks'.

Unfortunately, (CBR) encoding suffers [2] from some major drawbacks, like non-constant visual quality, and low coding efficiency. This is due to the fact that video may vary significantly from one frame to another, thus the quantisation [2] factors for macro-blocks among the different frames, will vary significantly, to compensate for the target constant bit-rate. As a result, [2] decoded video sequence exhibits inconsistent visual quality.

On the other hand, (VBR) encoding [2] can provide consistent visual quality, and higher coding efficiency for many video sequences, but it has very serious bit-rate and buffer-size constraints. Due to these limitations, there are few practical implementations of (VBR) encoder, especially for the transmission of digital video over 'IP communication systems', or the store of digital video on limited storage-space electronic devices. A free (VBR) encoder may not meet this requirement since the transient bit-rate may fluctuate significantly, thus provoking severe 'buffering' problems and 'start-up' delays, and of course, unpredicted storage file-size.

Unfortunately, previous work on the area seems not capable enough to solve all the problems, but only part of them. For example, in paper [6] a perceptually efficient (VBR) encoding algorithm was introduced, but this (VBR) encoder is unconstrained, and therefore, it cannot guarantee to satisfy fixed-size storage constraints. In paper [2], the authors used original frames for motion estimation and motion compensation. However, directly applying this scheme to low bit rate coding may cause significant quality degradation. This is because in low bit rate coding, there are significant errors between reconstructed frames and original frames. Finally, in paper [3], it is assumed that there is sufficient buffer size and pre-loading time, thus only the total bandwidth is considered, which is impractical for video streaming applications.

For this reason, a new approach that adopts two-pass variable-bit-rate (VBR) encoding to implement high-performance coding for video streaming and fixed storage applications is proposed.

Open Science Index, Electronics and Communication Engineering Vol:1, No:8, 2007 publications.waset.org/8876.pdf

In the first pass, the video sequence is, dynamically, split into segments/slices of random number of frames, while at the same time four (VBR) encoders will encode the video sequence and produce some statistics for each segment. This will help us to build an accurate 'R-Q function', so being able to obtain the exact relationship between the amounts of average bits/frame (R) produced for each segment and the quantisation factor (Q). By having this 'R-Q function', the quantisation factor for each segment could be optimized throughout the entire sequence, based on constraints such as decoder buffer size, transmission rate, or total storage size, before the 'second-pass' encoding is executed.

Finally, in the 'second-pass', the estimated quantisation factors for all segments will be used to encode the entire video sequence as a conventional (VBR) encoder. These statistics could be also used to encode multiple streams with (VBR) characteristics (Multi-stream VBR encoder). Because, the segments produced will be created in a 'predictable' and 'controllable' way, 'buffer underflow/overflow' could be also prevented under this scheme.

Experimental results based on an implementation of (H264) standard into FASTNETS platform ('Video Streaming Application' developed by BT), on a variety of test video sequences, have shown that the proposed (VBR) encoding scheme can provide more consistent visual quality and improved coding efficiency, while minimizing 'buffering' problems and controlling 'start-up' delay.

II. Main Algorithm

Under this scheme, a variable bit-rate video signal compression process will be described, in which the number of bits required to code a video slice, comprising of several video frames in uncompressed form is determined, and a quantisation level is selected for the transmission of the slice such that the overall bit rate of the slice corresponds to a predetermined value.

The video sequence is first divided into a plurality of slices, in each of which the number of bits required to code each frame falls within a range having a predetermined magnitude, and the quantisation level is determined for each slice

The need for two passes delays the transmission by the time taken to perform the additional pass. Like all digital encoding schemes, this arrangement relies on an adequate buffer store being available at the receiving end of the transmission, because the number of bits per frame varies, and it is necessary to store all the data necessary for the recreation of a frame before it can be generated.

In order to ensure that the receiver does not experience an underflow condition, the transmission rate of the bit-stream, at which individual segments of data are encoded, varies according to the amount of data required to generate each slice. Frames may be selectively omitted from transmission, in a controllable way, if the cumulative frame rate does not fall below a predetermined value (A), set at the beginning of an encoding session by the 'User'. The decoding process could be arranged to identify parts of the transmission from which frames may have been omitted, and to resynchronize the displayed stream. This may be done by extending the duration of individual frames, or by repeating some frames. Synchronization may be achieved by comparison between time stamps in the video stream and a corresponding audio stream.

A. MRC Function

In the first pass, the video sequence will be first analyzed by four (VBR) encoders, operating in parallel, to encode the video sequence for various quantisation levels. This step is performed on a 'frame-by-frame' basis. The processing power needed for this step is four times a standard (VBR) encoder. This 'R-Q function' is applicable to multiple streams of variable bit-rates, so it will be referred below as the 'Multi-stream Rate Control (MRC)' function. Two mathematical models will be used to determine (MRC):

MRC MODEL 1:

$$R = a' e^{-b'Q} \tag{1}$$

Q (30,45)
R: Average bits/frame
Q: Quantisation parameter
a', b': Modeling parameters to be determined.

The second model is a 3rd-order polynomial:

MRC MODEL 2:

$$R = aQ^3 + bQ^2 + cQ + d \tag{2}$$

Q [1, 30]
R: Average bits/frame
Q: Quantization parameter
a, b, c, d: Modeling parameters to be determined.

'MRC model 1' requires two modeling parameters (a', b') to be determined, and thus the statistics produced from two streams will be used to calculate the values of those parameters, while 'MRC model 2' requires four modeling parameters, and therefore all the statistics from four streams will be taken into account to determine their values.

Finally, on second-pass, the entire video sequence can be coded as in a conventional (VBR) encoder. The estimated quantisation factors (Q) for each slice will be used, determined by procedures described in the 'first-pass'. The resulting bit-stream is now ready to be transmitted to the receiving end, over any IP network. Finally, in 'second-pass', it is also possible to encode multiple-streams with (VBR) characteristics, based on the statistics produced on the first-pass.

III. Buffer Study

(VBR) inputs present a problem at the receiver side in ensuring that sufficient buffering resources are available.

Open Science Index, Electronics and Communication Engineering Vol:1, No:8, 2007 publications.waset.org/8876.pdf

There are two inter-related criteria to determine, namely the buffering capacity and the buffering delay. As the number of bits per frame varies, and the bit rate for the transmission channel is constant, the frame rate will vary, at the receiving end. The buffering delay required is that sufficient to allow the slowest frames (highest number of bits per frame) to be delivered and processed in time for them to be displayed, whilst the buffering capacity is determined by the capacity needed to store all encoded frames that have not been displayed yet.

In this part, a mathematical model will be expressed to describe the buffer level at the decoder side for 'Video Streaming' applications, based on 'buffer model' in (Fig. 1).

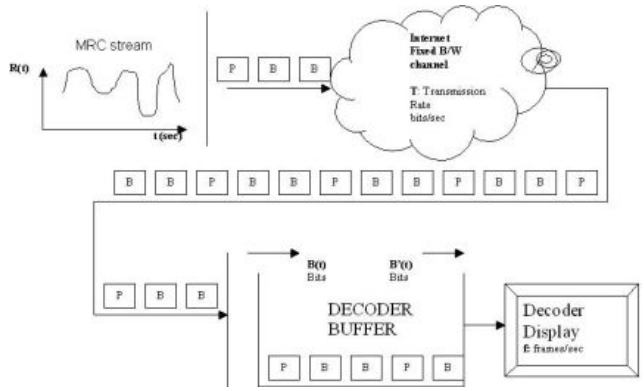

Fig. 1 The level of the buffer at the receiving end is illustrated, demonstrating how 'buffer level' is built up over time, when (VBR) stream is transmitted over a fixed bandwidth transmission channel

The number of bits (dB) contained in the buffer at a given time (t) is given by:

$$dB(t) = B(t) - B'(t) \tag{3}$$

B: Bits inserted into the buffer
B': Bits extracted from the buffer

Also, the number of bits (B) that were inserted into the buffer over a period of time (t) is given by:

$$B(t) = T * t \tag{4}$$

T: Transmission Rate (bits/sec)

Similarly, the number of bits (B') that have been extracted from the buffer over a period of time (t) is given by:

$$B'(t) = R(t) * f * t \tag{5}$$

R: Average bits/frame (MRC function)
f: Target frame rate (frames/sec)

Consequently, the number of bits (dB) remaining in the buffer at a given time instance (t) is given by:

$$\begin{aligned} dB(t) &= B(t) - B'(t) = \\ &= T * t - R(t) * f * t = \\ &= \left(T - R(t) * f\right) * t \end{aligned} \tag{6}$$

This function identifies the number of bits contained in the buffer at a given time instance (t), assuming that the transmission rate is ideal and fixed at a rate (T). Since transmission rate (T) and frame-rate (f) are predefined, the value of (dB) varies over time as a function of (R).

'Buffer underflow' may occur if the next frame is due to be decoded before the necessary data has arrived – in other words the buffer is empty. In order to avoid 'buffer underflow', it is usual to delay the start of the decoding process after the arrival of the first data in the buffer store. This introduces a delay in display of the video sequence to the end user, and it is desirable to minimize this 'start-up' delay.

From function (6) above, it is possible to determine a minimum value (dB_{min}), and also the time instance (t_{min}), at which that minimum value will occur. If this minimum value is negative, that is to say if there is a time (t_{min}) at which the number of bits received up to that point by the decoder is less than the number required to have been decoded to maintain the frame rate to the display device, a 'buffer underflow' condition will occur. A buffer delay is introduced (start-up delay), at the beginning of the decoding process for a period of time given by:

$$t_b = \frac{dB_{\min}(t_{\min})}{T} \tag{7}$$

t_b: Minimum 'start-up' delay (sec)

'Buffer overflow' may occur if there is not enough memory allocated for storing the video packets arriving at the receiving end. If the peak buffer size required could be determined before the transmission of the video sequence, sufficient buffer capacity could be reserved in the decoder in advance.

It has been already discussed that the number of bits (dB) contained in the buffer at a given time instance (t) is given by (6). By knowing the values of function (R), it is possible to identify a time instance (t_{max}), when property (dB) would reach its peak value (dB_{max}).

The buffer size (B_f) allocated to prevent 'buffer overflow' could be determined by:

$$B_f = dB_{\max}(t_{\max}) + |\, dB_{\min}(t_{\min})\,| \tag{8}$$

$|dB_{min}|$: Represents the absolute minimum value of bits that the buffer is primed to prevent 'buffer-underflow'.

dB_{max}: This is the peak value of bits contained in the buffer which would occur at a given time instance (t_{max}).

Open Science Index, Electronics and Communication Engineering Vol:1, No:8, 2007 publications.waset.org/8876.pdf

In free (VBR) transmission, the values of 'start-up delay (t_b)' and 'buffer memory (B_f)' cannot be predicted in advance, since they depend on the cumulative variable (R), which itself depends on the encoding process. However, the present scheme employs two-passes of the sequence at the encoder, so it is possible to use a buffer control process located in the encoder to determine the function (R), at the 'first-pass'. The parameters 'transmission-rate (T)' and 'frame-rate (f)' are also available at the encoder, so it is possible for the encoder to determine the required 'buffering time/start-up delay (t_b)' and 'buffer capacity (B_f)', during the 'second-pass' encoding and prior to the transmission of the (VBR) stream.

A. Buffer Underflow Prevention

The (VBR) stream would normally be transmitted over 'IP-network' with fixed guaranteed bandwidth (T). Recalling that the net number of bits (dB) remaining in the buffer at a given time instance (t) is given by (6):

$$dB(t) = \left(T - R(t) * f\right) * t$$

In order to avoid buffer underflow:

$$dB(t) \geq 0 \tag{9}$$

at any given time instance (t), throughout the entire sequence, and it follows that:

$$R(t) \leq \frac{T}{f} \tag{10}$$

To maintain function (R), below a value set by 'target transmission rate (T)', and 'frame-rate (f), some frames may have to be omitted prior to the transmission of the (VBR) bit-stream. In order to achieve this, the encoder could be controlled to selectively omit certain frames (B-frames) from the (VBR) stream. This can be performed in three different ways:

Firstly, it could be 'an off-line' process, which could take place after the termination of the '2-pass' (VBR) encoding session described earlier.

Alternatively, it could take place dynamically, during the 'second-pass' of the (VBR) process described earlier.

In a third possibility, the process could take place prior to the transmission of the (VBR) stream. This would be feasible if the 'Video Server' responsible for transmitting the (VBR) stream was checking the number of frames transmitted per second, and selectively dropping 'B-frames', if necessary, according to certain rules.

The number of bits transmitted over a period of time (t), must not exceed the number set by the target transmission rate (T). To achieve this, the number of bits produced over a period of time (t) is summed, and if necessary, 'B-frames' will be dropped, until the following condition is met:

$$\sum_{i=0}^{N} B(i) \leq T * t \tag{11}$$

$\sum B(i)$: Sum of bits produced over a period of time t.

T: Target Transmission rate (bits/sec).

N: Number of frames over a period (t).

'Buffer underflow' could be avoided by modifying the transmitted signal prior to the actual transmission of the (VBR) stream. A residual 'start-up delay (t_b)' could be allowed by the 'User', set prior to the start of the (VBR) encoding, such as fewer frames would be dropped, if necessary.

The overall process requires certain frames of the sequence to be dropped from transmission. Certain criteria need to be met to ensure that the consequent impairment of the quality is minimized. Referring back again to (MRC) function described earlier, it will be recalled that the 'sequence segmentation' process was arranged such that the variation in quantisation level within any given 'slice' was limited by a threshold parameter (A). In the process to be described below, this parameter is very important, as it will limit the 'frame-rate drop' to an acceptable level. This is always predefined prior to the beginning of a 'VBR encoding session'. The existence of this threshold ensures that the 'frame-rate' will not drop below a minimum value (f_{min}) given by (12):

$$f_{\min} = f * \left(1 - A\right) \tag{12}$$

f: Target frame rate (frames/sec)

A: Threshold given in (%).

The frames to be dropped can be only selected from the 'B-frames' alone, and the ratio of 'B-frames' over 'P-frames' should be given by (13):

$$\frac{N_B}{N_P} \geq k * A \tag{13}$$

N_B: Number of (B-frames)

N_p: Number of (P-Frames)

k : Constant selected to compensate for the relatively large size of (P-frames), comparing to (B-frames). Typically, this value is in the range of 1.5 to 2.

A: Threshold given in (%).

This ratio can be very easily defined in most implementations of the (H264) standard, prior to the start of (VBR) encoding.

B. Buffer Overflow Prevention

The process of preventing 'buffer overflow' is very similar to the buffer underflow prevention.

In order to avoid 'buffer overflow', the following equation should be true at any given instance t.

$$dB(t) \le M \quad (14)$$

M: Memory allocated at the receiving end (Bits).

The maximum transmission frame-rate at a given time instance t would be given by the following equation:

$$f_{max} = f * (1 + A) \quad (15)$$

f: Target frame-rate (frames/sec)
A: Threshold defined earlier in (%)
f_{max}: Maximum transmission frame-rate (frames/sec)

If condition (14) is met at any given time instance (t), there is no need to drop any frames, as 'buffer-overflow' will never occur. Memory should be large enough to store all extra frames.

But, in the case where this condition is not met, the decoder will start to drop 'B-frames' from the 'segment', which has been most recently delivered by the network. In that way, buffer overflow will never occur, while keeping video quality at a maximum level.

Finally, the receiver synchronizes the decoded video frame timestamps with those in the audio stream. It is obvious that no extra information will need to be transmitted prior to a 'video streaming session'. There is no degradation in the quality of individual video frames, but some of the perceived quality perception may be lost when the video will be displayed. This is due to the fact that a slight drop in the frame rate might occur, to compensate for 'buffer underflow/overflow' prevention. Nevertheless, the perceived quality will be significantly better and guaranteed comparing to the quality achieved by (CBR) encoding.

This scheme makes possible devices with limited memory space (mobile phones, PDAs etc) to be able to display a (VBR) stream as efficiently as possible.

IV. Test Results

To verify the effectiveness of the 'MRC function', some tests were performed demonstrating the quality of the video comparing to 'free VBR' and 'CBR' encoding.

In the first two graphs, the accuracy of the (MRC) function was tested for two given sequences.

In the next two graphs, (SNR) for (Y) component was measured and compared for (MRC), (VBR), and (CBR) encoding scheme. An implementation of the (H264) Standard into 'FASTNETS' platform was used to perform the tests. Two video sequences have been chosen, with the following type sequence and random number of frames:

A. 'BBC news' sample (QCIF, 320 frames). The picture type is: IBPBPBPBPBPBP…

This is a relatively video static content, with a few number of scene changes.

B. 'Football' sample (QCIF, 215 frames). The picture type is: IBPBPBPBPBPBP…

This is a high-motion video clip, with one large scene change occurring near the end of the clip.

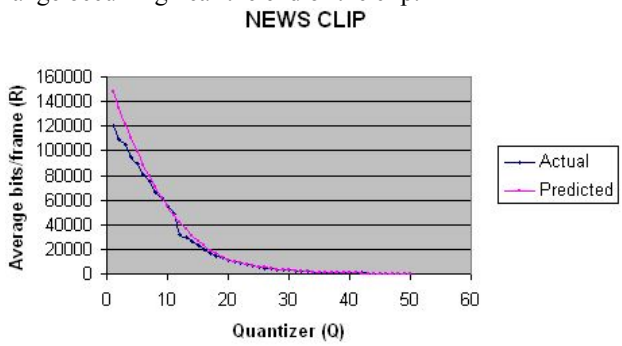

Fig. 2 This graph demonstrates that MRC function (R) is very close to the 'actual'-measured (R), for 'quantizer (Q)' ranging from [1,44]. Maximum deviation from the predicted one occurred for (Q=12), at a percentage of (22%)

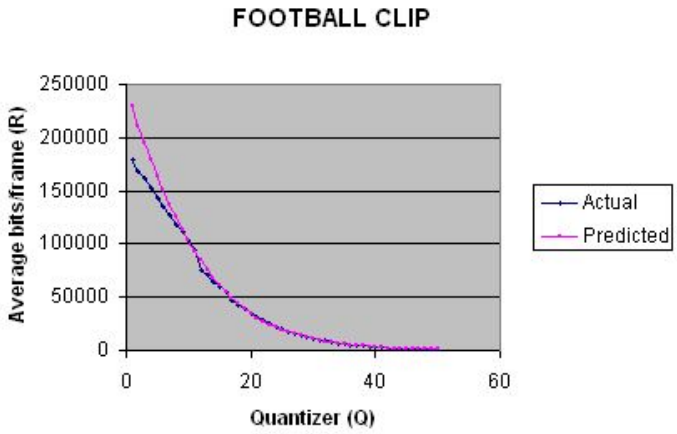

Fig. 3 This graph demonstrates that function (R) has also a very good match with the 'actual'-measured (R), for 'quantizer (Q)' ranging from [1,44]. Maximum deviation from the predicted one occurred for (Q=1), at a percentage of (22%)

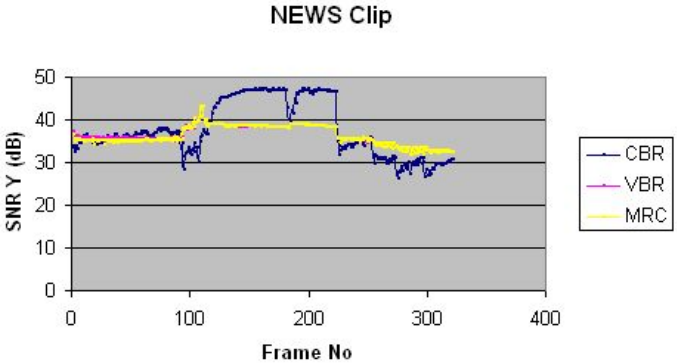

Fig. 4 This graph demonstrates (SNR) for (Y) component, for 'BBC News sample'. Maximum deviation of (SNR) between adjacent frames: (CBR=11.61dB, VBR=3.61dB, MRC=3.70dB)

Open Science Index, Electronics and Communication Engineering Vol:1, No:8, 2007 publications.waset.org/8876.pdf

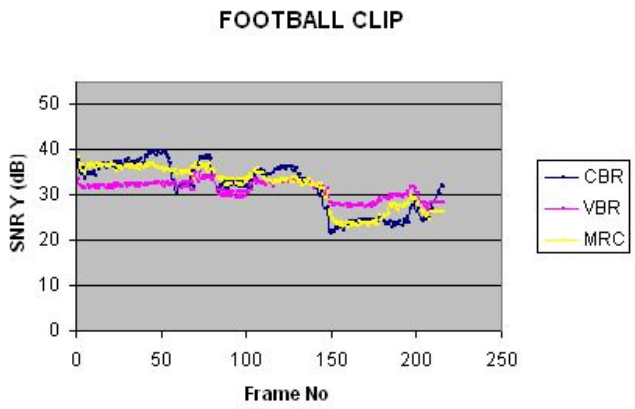

Fig. 5 This graph demonstrates (SNR) for (Y) component, for 'Football sample'. Maximum deviation of (SNR) between adjacent frames: (CBR=5.15dB, VBR=3.91dB, MRC=4.47dB)

It is obvious that in 'Fig. 3&4', (MRC) can keep the 'good characteristics', which can be found in '(VBR) encoding scheme' (almost constant video quality), while at the same time, network transmission could be kept friendly to 'buffer limitations', like a (CBR) encoding scheme. Note that (MRC) content was coded with a threshold deviation (A=30%), in other words, frame-rate will drop at a maximum value of 30%, if necessary, to avoid 'buffer underflow/overflow'. This would be easily achieved by dropping (B-frames).

V. Conclusion

Summarizing, it was clearly demonstrated that a (VBR) bit-stream could be delivered to a variety of devices, connected to a number of IP networks (fixed/mobile). All such devices would perceive the same video quality regarding 'video frame quality', but lower bandwidth devices would experience a reduced 'frame-rate'. In each receiving end device, there will be the option of setting up parameters like 'target transmission rate (T)', 'target frame-rate (f)', 'start-up' delay, etc, prior to the start of a (VBR) encoding session. This parameters could be set according to the limitations of their given connection, thus producing a (VBR) friendly and 'buffering efficient' bit-stream. This method offers great flexibility comparing to 'pure (VBR)' or (CBR) encoding. In 'unconstrained (VBR)', there are serious buffer constraints, while in (CBR), the need for creating more than one stream for different bandwidth connections, and the limitation of poor video (non–constant) quality, could reduce significantly the overall video quality perception. The scheme described in this paper offers great flexibility as it combines the best characteristics from 'two different worlds' (CBR and VBR), in a very efficient way.

References

[1] Kai Sun, Mohammed Ghanbari, Ian Henning, Matthew Walker, Othon Kamariotis "A stored VBR video transmission scheme over Internet", PacketVideo 2003

[2] Yue Yu, Jian Zhou ,Yiliang Wang,Chang Wen Chen. "A novel two-pass VBR coding algorithm for fixed-size storage application", Circuits and Systems for Video Technology, IEEE Transactions on Publication Date: March 2001 On page(s): 345 - 356, Volume: 11, Issue: 3, ISSN: 1051-8215, Reference Cited: 16

[3] Jianfei Cai , Chang Wen Chen. "Optimal bit allocation for low bit rate video streaming applications", Image Processing, 2002. Proceedings. 2002 International Conference on Publication Date: 2002,On page(s): I-73- I-76 vol.1, Volume: 1, ISSN: 1522-4880, Number of Pages: 3 vol (lxx+991+976+1008)

[4] PH Westerink, R. Rajagopalan and CA Gonzales. " Two-pass MPEG-2 variable-bit-rate encoding", IBM Journal of Research and Development, vol.43, no.4, July 1999 pp pp 471-88

[5] N. Mohsenian, R. Rajagopalan, C. A. Gonzales. "Single-pass constant- and variable-bit-rate MPEG-2 video compression", IBM Journal of Research and Development, vol.43, no.4, July 1999 pp pp 489.

[6] M. R. Pickering and J. F. Arnold, "A perceptually efficient VBR rate control algorithm," *IEEE Trans. Image Processing*, vol. 3, pp. 527–532, Sept. 1994

[7] Liew, S.C.; Tse, D.C.-Y. "A control-theoretic approach to adapting VBR compressed video for transport over a CBR communications channel" Networking, IEEE/ACM Transactions on, Volume: 6, Issue: 1, Feb.1998, Pages: 42-55.

Open Science Index, Electronics and Communication Engineering Vol:1, No:8, 2007 publications.waset.org/8876.pdf