Review of the Challenges to Remove Jitter and Packet Loss during Continuous Playback of Streamed Video Data in Video on Demand (VOD) System Receivers

Dr. Rajesh Bhadada
Dept. of Electronics and Communication

N C Barwar
Dept of Computer Science & Engg.

Rajendra Bhansali
Dept. of Electronics and Communication

M.B.M. Engineering College, J.N.V. University, Jodhpur 342 011, India

ABSTRACT
Video on Demand (VOD) system is inherently resource intensive and demanding in respect of performance for continuous playback due to bulk size of video data, length of video session and real time continuous playback. The data stream is to be played back strictly in original sequence for longer periods in general once playback begins. Any interruption in data transfer stream, unordered delivery of packets or loss of packets can result in jitters or breaks during video playback. In practice the ordered delivery of data packets through public domain networks is not guaranteed, which can break the sequential notion of video data. These are the main challenges for jitter and breaks free VOD services. This paper, presents an in-depth reviews of techniques for removal of jitter and to make up against packet loss. Fixed or adaptively delayed play back may be useful for removal of jitter during video playback. Techniques called forward Error Correction, Interleaving and Interpolation are used to fight against the loss of packets, which have also been examined critically. These techniques are an important part of media players (VOD receivers) designs and streaming servers.

Indexing terms: VOD, packet loss, multimedia, video playback, jitter

1. INTRODUCTION

An on-line video transfer system has a huge storage of video contents at the servers which may be selectively transferred on demand. These video data are not only bulky, but also needed strictly in original sequence during playback. The playback ranges from a few minutes and goes for few hours commonly.

A server has to handle an individual's request for a video with a single unicast or a multicast stream in the form of video data stream to the user's media player. In practice the ordered delivery of data packets through public domain networks is not guaranteed, which can break the sequential notion for playback of streamed video data. Congestion control strategy in modern networks tends to lower down the transmit rate for some period after packet loss leading to further deterioration of conditions for smooth playback [7,13]. User's VCR like interaction such as pause, FF, REW, Slow Forward, Slow Reverse, Jump Forward, Jump Backward cause additional set-up load at the server and can cause jitters, jams or breaks in video during playback [16].

The video data can be transferred in one of the following manners -

(i) For instantaneous or live video data transfer, the data are sent at a constant rate x(t) equal to the consumption rate d(t) at the receiver, where x(t) has to be at least equal to the video encoding rate. The server clocks out the compressed video via the network. User equipment decompresses the video/audio and plays it back, immediately after receipt. Ideally, no initial wait or buffering is required. But due to varying conditions in real networks, it is nearly impossible to download continuously in original sequence of transmission [9,19]. This results in jerky and jammed video replay.

(ii) Under practical network conditions to achieve smooth playback, the user VOD player (receiver) delays playback for a short period of few seconds to eliminate network induced jitter. The receiver does so by first storing the compressed video received from the network into a buffer. Once the receiver has prefetched a few seconds of the video, it begins to play. In the meantime, receiver and the network have the opportunity for recovery or retransmission of lost packets [14].

This paper presents review of techniques and measures to achieve jitter and break free play back of video from a streaming server. Considerations and techniques for removal of jitter during playback have been provided in section 2. To make up against packet losses to maintain strictly sequential data for original playback (section 3), techniques such as Forward Error Correction (section 3.1), interleaving (section 3.2) and Receiver Interpolation (section 3.3) have been reviewed. The paper concludes by summarising the conditions wherein these techniques could be more successfully applied.

2. JITTER REMOVAL DURING PLAYBACK

The video player (VOD receiver) is expected to provide synchronous and smooth playback of video chunks. Thus it needs to counter inconsistent conditions in network which cause variable data transfer rates, resulting in induction of random jitters during playback. Modern packet switched store and forward networks do not guarantee ordered packet delivery. Hence it is mandatory for the receiver to be able to rearrange packets in original order before playback, to maintain continuity of correct playback and to detect any packet loss. Packets are considered lost and forgotten if they fail to reach before their scheduled playback time [12, 15].

The following three mechanisms when applied in combination can eliminate or facilitate removal of the jitter [2,22] -

1. Allotment of a sequence number to each playable video data chunk, which is incremented by one for each of the packets generated/sent by the source in conformance of original order of the video.

2. Attachment of a time stamp with each data chunk, where the time stamp is the time at which data chunk was generated / sent by the source. Thus the packets get ordered implicitly by increasing time stamp.
3. Delayed playback of data chunks by the VOD receiver by such smallest duration so that almost all the packets are received before their playback times. This delay can be fixed or may vary adaptively during the video session.

Fixed Delay playback and Adaptive Delay playback are detailed as below:

Fixed Delay Playback

In this strategy, if a video packet is time stamped at *x*, then the receiver plays it back only at time *x+q*. Such a packet has to be arrived by that time else it needs be discarded and considered lost. In other words the receiver plays back each packet exactly after *q* units of time since the packet was generated. Smaller the *q*, more satisfied will be the user [4]. When an attempt is made to reduce *q*, there is an increase in the number of packets that fail to reach before scheduled time of playback. That is the risk that many packets may miss scheduled playback time increases and subsequent increase in jitter will be experienced. As a thumb rule, it is preferable to use large *q*, if the large variation in end-to-end delay is common; else if when delay and the variations in such delay are small, keeping *q* small may be more efficient. Fig. 1 depicts the limits for the consideration of packet loss vis-à-vis playback delay. The time at which packets are generated and played back for a single session are shown. Cases of two distinct initial playback delays are taken at time p, p' (p' > p) on the time scale. Packets generated by the server at regular intervals are shown as the leftmost staircase. The first packet arrives at time *r* at VOD receiver (player). Due to varying conditions at the network, subsequent packets are not guaranteed to reach with same delay. It is shown by uneven spacing (shown by dotted line) amongst received packets. In case of playback schedule marked "A", the fixed playback delay is set to *p-r*. It is evident that at least two packets miss the deadline of arrival for their scheduled playback, leading to jerk or jam during playback. Thus they are to be considered lost. Instead for the case of playback schedule "B", the fixed playback delay is set as *p'-r* (where p' > p), it is clearly observed that all packets have arrived before their scheduled playback times, therefore no loss. It is to imply that keeping fixed delays larger results in better jerk and break free playback performance by the receiver. The larger delay however is difficult to be accepted by the users, who would not prefer to wait longer for the start of playback for their demanded video.

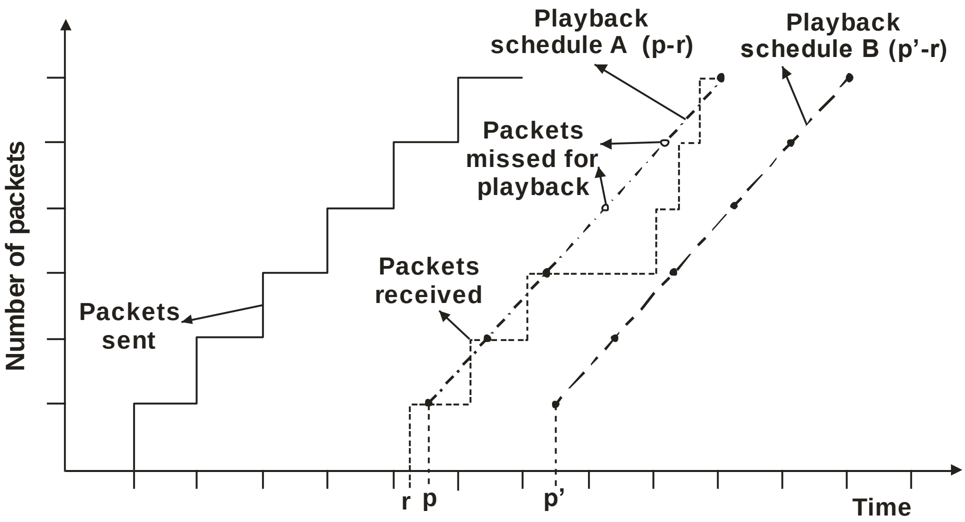

Figure 1: Impact of different fixed playback delays over packets loss

Adaptive Delay Playback

For a loss less playback performance, the fixed delay playback strategy favours longer initial delay in playback. The larger delay enables more (most) packets to reach the VOD player before their deadlines and thus negligible loss. Human users get bored in cases of larger initial delays and have tendency to change their minds (termed reneging) i.e. they tend to change "channels" or even switch off [4]. The situation worsen with VCR like interactions performed by the users. In simple words, long delays can become intolerable.

Ideally, the playback delay must be minimized, i.e. receiver design should be able to tolerate or tackle a loss of packets to some extent (say below a few percent limit). Due to varying conditions, networks can not operate at the same levels causing variable delay. Thus an optimization in the initial delay for the playback with respect to the estimated network delay and its variance will result in more bearable performance. Accordingly an adjustment in the playback delay in the beginning and during the video session can achieve more user satisfaction.

A VOD player with the information of timestamps can estimate to adapt/adjust its playback delays to minimize it.

Table – 1 exhibits important parameters and their notations.

Table – 1: Notations

Symbol	Description
t_i	the timestamp, when the packet was sent/generated
r_i	the time when packet *i* arrives at receiver
p_i	the time when packet *i* is played back at player
d_i	average network delay for the *ith* packet
v_i	average deviation of the delay from d_i.

The difference r_i-t_i represents end-to-end network delay for the i^{th} packet. It may keep varying packet to packet under changing conditions in the network.

The average network delay d_i for receiving the i^{th} packet can be estimated from the timestamps by:

$$d_i = (1\text{-u})\, d_{i-1} + \text{u}\,(r_i - t_i) \quad \ldots(1)$$

where u is a fixed small percent like constant (say 0.01) [6]. Thus d_i represents a smoothened average for the observed network delays $(r_1 - t_1)$, $(r_2 - t_2)$… $(r_i - t_i)$. This estimate considers more weight on the network delay caused to packet received most recently by including d_{i-1} as compared to observed network delays of packets received earlier.

The v_i can then be estimated from the timestamps as-

$$v_i = (1\text{-u})\, v_{i-1} + \text{u}\,|(r_i - t_i - d_i)| \quad \ldots(2)$$

The estimates d_i and v_i can be calculated for every packet received. It is then used to set the time point for playback for the first packet for the next video session. After estimation, the receiver sets playback delay. If packet i is the first packet, its playback time p_i can be set as:

$$p_i = t_i + d_i + \text{K}v_i \quad \ldots(3)$$

where K is some positive constant (say K=4) [6]. The term Kvi offsets (shifts) the playback time so that only a small number of the arriving packets through the streamed video data are lost. After applying an offset from the point in time when the first packet was played back, the playback point for any subsequent packet can be predicted. If q_i represents the difference $p_i - t_i$ (i.e. the length of time since the first packet is sent/generated to till its playback instant) and for a packet j which also belongs to the same video chunk, then packet j will be played back from the instant $p_j = t_j + q_i$. The receiver can find whether a packet is the first packet in the video chunk comparing the timestamp of the i^{th} packet with that of the $(i\text{-}1)^{st}$ packet. If $t_i - t_{i-1} > x$ (where x is the size of a preset video unit playback time), the receiver can make out that the i^{th} packet starts a new video chunk. In case of packet(s) loss, two successive packets received may have timestamps difference $> x$, or the two packets still belong to the same chunk. It can be resolved by use of sequence numbers by the receiver, to determine if the difference in time stamps $> x$ is on account of a new video chunk or due to lost packets.

3. RECOVERY AGAINST PACKET LOSS

The techniques discussed in the last section can be applied to counter variable delays caused by network to result in jitter. But if there are packet losses, there is strong possibility of break in maintaining continuity in playback. Thus consistent playback is the next challenge for the VOD player. As explained earlier that in case of a video playback, a packet is declared lost if it never reaches the receiver or it arrives after its playback time. Lost packets could be makeup by their retransmission or could be recovered by use of loss recovery methods.

Loss recovery schemes attempt to preserve acceptable video quality against packet loss. Retransmitting lost packets is generally cumbersome and bulky in a real time streaming type video transfer via a wide area network. Retransmission of such a packet that has already missed its playback deadline renders whole retransmit process useless. If packet(s) get stuck due to overflow at a router queue, they can not be retransmitted quickly enough normally [25]. Special techniques are to be applied for maximal recovery against packet losses. Forward error correction (FEC) or Interleaving techniques are applicable at the server, while the Interpolation technique is applicable at the receiver (VOD player) to recover/maintain video playback in anticipation of packet losses.

Forward Error Correction (FEC)

Under this strategy, redundant information in one form or other is added to the original packet stream. Such redundancy may be used to reconstruct approximate/exact version against some of the lost packets. However redundancy causes direct loss due to its overhead either at storage or at transmission. To neutralize the effect of such an overhead, the transmission rate of the stream can be marginally increased.

There can be two types of FEC mechanisms:

- A redundant packet is obtained by Ex-OR of the n original packets of video data and that redundant packet is sent in succession after these n packets to form one chunk of video data. With such a stream of video data, if any one packet from the chunk (containing group of $n + 1$ packets) is lost, the receiver can fully construct the lost packet. The receiver fails to reconstruct if there is loss of two or more packets in one chunk (group). By keeping n (group size) small, most of lost packets can be recovered when the loss is not excessive. To counter the effect of redundant packet, the stream transmission rate needs be increased by a factor of $1/n$. Thus, this strategy is not favourable if the group size is having smaller number of packets. Say if $n = 3$, then the transmission rate needs be increased by about 33% which may be difficult to achieve with an existing network setup. The increased transmission rate can also deteriorate signal quality; hence more errors may creep in. Due to addition of redundant packet, the playback delay also tend to increase, as now the receiver is required to wait to receive the entire group of packets including redundant ones before it can begin playback.
- In another class of FEC, in place of above a lower resolution video stream is sent as the redundant information. The server has to create a normal video stream as well as its corresponding low resolution low bit-rate video stream. The low bit rate stream acts as redundant stream. The sender constructs the n^{th} packet by taking the n^{th} chunk from the normal stream and appending it to the $(n\text{-}1)^{st}$ chunk from the redundant stream (Fig. 2). In case when there is a non-consecutive packet loss, the receiver can conceal the loss by playing back the low bit-rate encoded chunk that arrives with the succeeding packet. This low bit rate chunk produces lower quality video in comparison to the normal chunk. Such occasional low quality video in place of broken/jammed playback, to continue the video (of mostly high quality chunks), gives overall better video presentation. The advantage in use of this technique is that the receiver needs to receive only two packets before playback, resulting in smaller increase in playback delay. Further, due to low bit rate encoding of redundant chunk, there shall be only marginal increase in the required transmission rate.

To cope with consecutive losses, a simple variation can be applied. Instead of appending just the $(n\text{-}1)^{st}$ low bit rate chunk to the nth normal chunk, the $(n\text{-}1)^{st}$ and $(n\text{-}2)^{nd}$ or $(n\text{-}1)^{st}$ and $(n\text{-}3)^{rd}$ and so on can be appended. Such a variation wherein more number of low bit rate (preceding) chunks are appended to each normal chunk, the video quality can become acceptable for a wider variety of losses during long sessions of continuous video playback. However the drawback is that the additional chunks increase the transmission bandwidth and the playback delay.

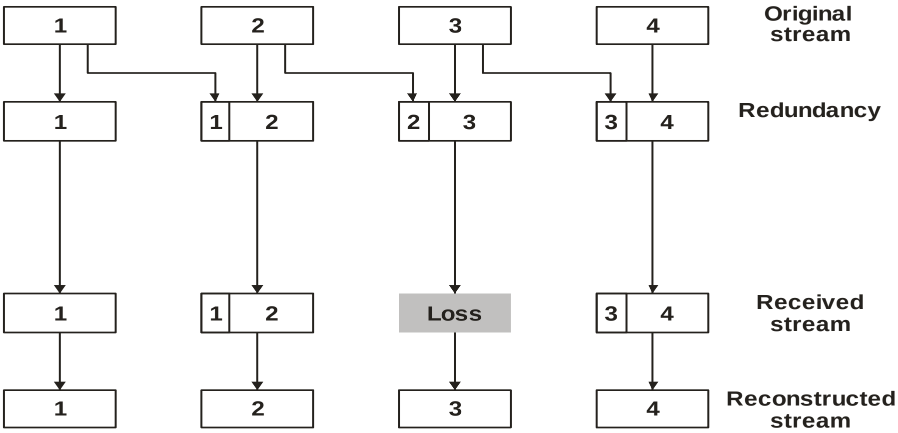

Figure 2: Lower quality redundant information piggybacking

Interleaving

Instead of sending redundant transmission, interleaved data packets are transmitted in this technique. First, at the server a resequencing of units of video data is done before transmission (Fig. 3). As depicted, originally adjacent video data units have been separated by a certain distance in the transmitted stream. Say (as illustrated in fig. 3, but could be generalised) if a packet unit is 5 mSec of playable length and while one chunk of video playable unit is 20 mSec (i.e. four units per chunk), then the first chunk is resequenced to contain units 1, 5, 9, and 13, while the second is resequenced to contain 2, 6, 10, and 14 and so on. The loss of a single chunk from an interleaved stream results in multiple small gaps in the reconstructed stream; otherwise there would have been a large single gap which could cause a break.

Such simple interleaving can bring in significantly improved perceived quality of a video stream and it is burdening only a low overhead. The drawback is that it increases latency. This can limit its use for interactive real time video, but it can perform well for streaming stored video. Other main advantage offered is no increase in the bandwidth requirements.

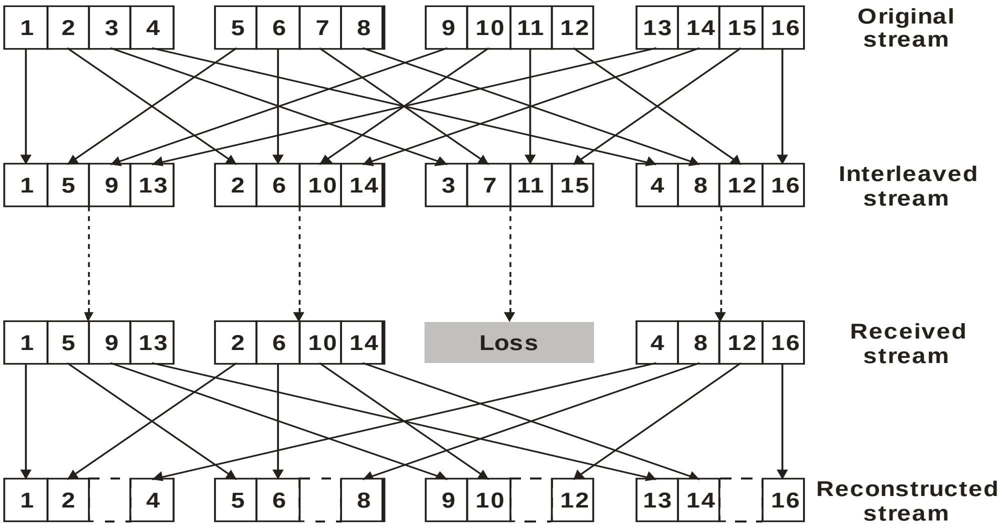

Figure 3: Sending interleaved (resequenced) data packets

Receiver Interpolation Scheme:

This technique is applied at the receiver against packet losses. Interpolation of data is done by the receiver, to attempt to hide the packet loss, which could result into jam or break in video playback. Such interpolation schemes make receiver efficient to produce a replacement for lost packets that are similar to the original. The video data, in particular the motion video has large amounts of short term self-similarity. The packed repetition is the simplest form of receiver based recovery. A receiver can simply interpolate by replacing for lost packets with copies of the packets that are immediately proceeds in the playback sequence. Such a technique has low computational complexity, simple to implement and performs reasonably well.

A real interpolation can be applied by the receiver to attempt recovery by use of an extended interpolation, which uses audio and video before and after the loss to reconstruct a suitable packet to cover the loss. Interpolation performs better than simple packet repetition but is significantly more computation intensive. Such computations hamper and make difficult to meet real time playback performance [15,22] during video playback. Such techniques could yield reasonable results for relatively small loss rates (<10 percent), and for small playback length packets (4-40 mSec). When the loss length goes above the length of a few frames (say 3 to 5 out of 30 frames for a moving video) these techniques break down, as jerks or stills will be seen.

4. CONCLUSION

For streamed video transmission under the system such as VOD, to counter or eliminate the effect of network jitter Sequence Numbers, Timestamps and Delayed Playback can be applied. To counter or recover from the video jamming or breaks during playback caused by packet losses the techniques of FEC and Interleaving are applicable at the server, while the Interpolation can be incorporated in the VOD player (receiver). However there could be a significant reconsideration when there is streaming of live real-time interactive video instead of streaming stored video. Streaming of stored video can tolerate significantly larger delays. When a user requests a video, he/she may accept and tolerate to wait few seconds or more before playback begins, as well most users can tolerate similar delays after interactive actions such as a jump forward or backward. This greater tolerance for delay gives the application developer greater flexibility when designing for stored video applications as compared to real time live video streaming.

5. REFERENCES

[1] P. Venkat Rangan and Harrick M. Vin, "Efficient Storage Techniques for Digital Continuous Multimedia", IEEE Transactions on Knowledge and Data Engineering Vol. 5, No. 4, pp 564-573, August, 1993.

[2] Phillip Lougher, Doug shepherd and David Pegler, "The Impact of Digital Audio and Video on High Speed Storage", Thirteenth IEEE Symposium on Mass Storage Systems, pp 84-88, 1994.

[3] Hari Kalva and Borko Furht, "Techniques for Improving the capacity of video on demand systems", Proceedings of 29th International Conference on Systems Sciences, IEEE, pp 308-315, 1996.

[4] Wu, Ma and Lin, "On queuing with customer impatience until the beginning of service", IEEE proc. Computer Performance and Dependability, 1996.

[5] Edward Chang and Hector Garcia, "Reducing Latency in Media Servers", IEEE Multimedia, Vol. 4, No. 3, pp 50-61, 1997.

[6] Marlha L. Escobar and Shahram Ghandeharizadeh, "On Co-ordinated Display of Structured Video", IEEE Multimedia, Vol. 4, No. 3, pp 62-75, 1997.

[7] T.R. Dua, "Multimedia Infrastructure Requirements Issues, Challenges and Opportunities", Electronics Information and Planning, Vol-25, No. 6, March 1998.

[8] H O Srivastava, "Broadcast Technology – Past, Present and Future : A Review", IETE Technical Review, Vol. 16, Nos. 3 & 4, pp 317-334, May-August 1999.

[9] Derek Eager, Mary Vernon and John Zahrojan, "Bandwidth Skimming: A Technique for Cost Effective Video on demand", IEE proceedings of Multimedia Computing and Networking 2000, pp 1-10, San Jose CA, Jan 2000.

[10] Gary Chan, "Trade-off between system profit and user delay/loss in providing near video-on-demand service", IEEE transactions circuits and systems for video technology, vol 11, pp 916-926, Aug. 2001.

[11] Eenjun Hwang, B. Prabhakaran and V. S. Subrahmanian, "Presentation Planning for Distributed VOD Systems", IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 5, September / October 2002.

[12] Paolo Bellavista, Antonio Corradi, "How to support Internet based distribution of video on demand to portable users", proceedings of 7th IEEE International Symposium on Computers and Communication ISCC '02, IEEE Computer Society, pp 1530-1546, 2002.

[13] W F Poon, K T Lo and J Feng, "Determination of efficient transmission scheme for video on demand services", IEEE transactions on circuits and systems for video technology, vol 13, pp 188-192, Feb 2003.

[14] Shiwen Mao et al, "Video transport over ad-hoc networks : Multistream coding with multipath transport" IEEE journal on Selected Areas in Communications, pp 1721-1737, December 2003.

[15] Ofer Hadar, Merav Huber, Revital Huber and Raon Shmueli, "Quality Measurements for Compressed Video Transmitted over a Lossy Packet Network", Society of Photo-optical Instrumentation Engineers, Vol. 43, No. 2, February 2004.

[16] W F Poon and K T Lo, "Delay requirement for transmission of VBR videos in interactive broadcast environment", Proceeds of 2nd international conference on Information Technology for Application ICITA, pp68-73, 2004.

[17] Thinh Nguyen and Avideh Zakhor, "Multiple Sender Distributed Video Streaming", IEEE Transactions on Multimedia, pp 315-326, April 2004.

[18] Rong Yan and Alexander G. Hauptman, "Co-Retrieval : A Boosted Reranking Approach for Video Retrieval", International Conference on Image and Video Retrieval, Dublin City University, Ireland, July 21-23, 2004.

[19] Kien A. Hua, Mounir A. Tantaoui and Walla Pak Tavanapong, "Video Delivery Technologies for Large Scale Deployment of Multimedia Applications", Proceedings of the IEEE Vol. 92, No. 9, pp 1439-1448, September 2004.

[20] Bhadada Rajesh & Juneja RBS, "Contemporary issues in secured information recording and replay of digital video sequences on demand from users", Proceedings of National Conference on Current Advances in Information Protection and Security CAIPS-05 of AICTE, CSE Dept, MBM Engg. College, JNV Univ. Jodhpur, pp 89-99, Feb. 26-28, 2005

[21] Bhadada Rajesh and Sharma Dr. K L, "Cost optimization in VOD systems for efficient information management", Proceedings of National Conference on Emerging Computational Techniques and Their Applications ECCTA-05 of UGC, EE Dept, MBM Engg. College, JNV Univ. Jodhpur, pp 264-269, Oct. 22-23, 2005

[22] Jun Peng, Biplab Sikdar, "An Efficient and scalable loss recovery scheme for video multicast", IEEE Transactions on Multimedia, pp 356-365, April 2005.

[23] Bhadada R. and Juneja RBS, "Function Oriented Design Considerations in VOD servers", Proceedings of National Conference on Emerging Trends in Electrical Electronics and Communication Engineering, ECE and EE Dept, Shaheed Udham Singh Engg. College, Tangori, Mohali, Punjab, pp 178-181, Nov. 10, 2006. Bhadada R. and Sharma Dr. K L, "Analysing Physical Storage allocation schemes for multimedia applications", Proceedings of National workshop on Multimedia Animation RGVP Univ., CSE Dept, Govt. Engg. College, Ujjain, pp 49-53, April 16-17, 2007.

[24] Bhadada R., "Study Of Efficient Information Management For Video On Demand Systems" Ph.D. thesis under supervision of Dr. Sharma K.L., ECE Dept., MBM Engg. College, JNV Univ. Jodhpur, Jan. 2008.

[25] Rai B. and Bhadada R., "Design and performance estimation of Multimedia Information Distribution", Proceedings of International Conference on Emerging and Futuristic System and Technology, pp 150-156, L.I.E.T., Alwar and Institution of Engineers (India), April 09-11, 2009