

Video Streaming in Peer-to-Peer Networks Using Network Coding Renders Efficient Video Cassette Recorder Operations

¹Arockia Xavier Annie R., ¹K.V. Pradeepthi and ²P. Yogesh

¹Department of Computer Science and Engineering, Anna University, Chennai, India

²Department of Information Science and Technology, Anna University, Chennai, India

Abstract: Problem statement: Major technological development in recent years has led to the usage of shared streaming solutions by Peer-to-Peer (P2P) Video-on-Demand (VoD) systems in the Internet. Video streaming in P2P network systems has ample amount of loss of video packets due to network disabilities such as congestion, intrusion, connectivity problems, excessive network collisions etc. Adding onto the video streaming problems, Video Cassette Recorder (VCR) operations require flexibility of playback to the user. So, any missing or randomized packets from the video have to be instantaneously corrected and generated appropriately. **Approach:** In this paper, we study the working of VoD streaming system that uses Network Coding (NC) for improving the delivered video content at the end-user by correcting the error packets. We, study that NC not only, materializes uninterrupted playback but efficiency in VCR operations particularly, the seek operation have also improved the user perceived quality of videos. Our setup handles the NC generator present at the proxy between the media server and the peer clients, reducing the overhead at the server. The relevant packets that are lost within each peer-client are generated with the NC packets. **Results:** The receiver detects error due to loss of packets and corrects at a much faster pace than the time consumed for retransmission. This helps in improving user efficiency in VCR operations also. Though, NC provides added advantage in P2P VoD systems, there is initial transmission delay, a time cost incurred in video streaming. This time cost is rather small when compared to the difficulties within the Internet for the retransmissions. **Conclusion:** From the study we observe, that NC when applied to P2P VoD has few difficulties. They are complexity in implementing NC and tradeoffs on the part of NC in video streaming. Based on time and cache constraints these difficulties are not overwhelming when the actual benefits reaped are for a longer period of time.

Key words: Video-on demand, peer-to-peer systems, network coding, VCR (random seek) operations

INTRODUCTION

Video streaming has become a common day to day activity for all internet users. Recent advances in multicast video streaming algorithms have opened up new ways to benefit Video-on-Demand (VoD) services to potentially millions of users with the help of Peer-to-Peer (P2P) network groups. Peer-to-Peer Video-on-Demand system (P2VoD) is a distributed network system, where the central control of the system is not given to any fixed node. Every peer in the system acts as a server and client put together. Due to control decentralization at peer networks, failure nodes do not affect the system in such a way that it would lead to network failure (Thomos and Frossard, 2009b). Though, P2P has proven to be a fitting success when it comes to serving streaming requests, it still has major drawbacks when it comes to serving users without corrupt or lost packets. This creates message traffic and retransmission traffic at the peer nodes. Also, this brings down the quality of the streamed content such as

video in the case of VoD. If a few packets are missed in data transmission, then the results are to some extent tolerable, but in video, even if some packets are corrupt, the output is not tolerable as it is highly delay sensitive. In order to provide quality content for high profiled users who pay more we bring out a solution using Network Coding (NC) concept which would lead to efficient and fast recovery (Chan *et al.*, 2010). The theoretical study in math has provided a vast advantage of using Network Code over the other preferred (non NC) multicast streaming mechanisms (Li *et al.*, 2003). Though it is complex to implement in real world scenario, this mechanism proves to be more advantageous than most other systems (Nguyen *et al.*, 2007).

Using the concept of Network Coding, packet loss and errors during packet transmission can be rectified with ease and provide efficient content when compared with regeneration of packets through other error correction methods such as FEC, etc (Yu *et al.*, 2007). In this paper, we setup a video streaming system in a Peer-to-

Peer environment with a proxy between the Media Server and the peers. To provide with maximum video efficiency, throughput, error resilience and adaptability, the system is designed to withstand resilience and fault tolerant. And Network Coding has been used to achieve the enhancement at the user end with little drawback on the initial playback latency.

In P2P networks, the distribution of the video is done through packets. When a file is downloaded by many users, these users help each other so that the server load is significantly reduced. When users are dynamic, few packets are tent to go missing from the network. Network Coding helps in packet availability. Network Coding is a concept where along with the original message, some more packets are piggybacked. Any lost message packets can be retrieved from these piggybacked packets. The nodes do not relay the packets that they receive as it is, they will combine the packets and send them out as the piggybacked packets along with the original message. When a video is being transmitted from source to destination, the video is split into packets. When the node is transmitting packets to other nodes, all the packets do not necessarily travel through the same channel. So each packet might end up travelling in a different channel. When they reach the destination, the quality of the video packets that has been received has to be ascertained. If any packets are lost in the transmission, then re-transmission of these packets causes delay in play back of the video or causes jittered video play. This is where the Network Coding comes in handy. Every peer node has a network coder in it. When transmitting packets, it performs coding on all the packets that are present with the node at that point and piggyback them along with the video packets. At the destination end, if there are any lost packets, then the packets are regenerated using these piggybacked packets.

There are two types of Network Coding, linear and random. In Linear Network Coding, each node generates a new packet which is a linear combination of packets received by it, as explained in (Thomos and Frossard, 2009a; Li *et al.*, 2003; Ho *et al.*, 2003 and Wang and Li, 2007). In Random Network Coding, each node independently chooses mapping of input to output to generate Network Coded packets as given in Niu and Li, 2007; Ho *et al.*, 2003. The linear combination of the input packets, with randomly chosen coding co-efficient is done in our work. The working of NC in P2P networks and its development from the normal networking has been given in Huang and Zhang, 2011. We use the Linear Network Coding technique, as it is simpler to implement linear encoder and decoder in practice (Li *et al.*, 2003). Random Network Coding takes more time to encode and decode

as the complexity is more. Linear Network Coding helps to achieve more throughputs in multicast networks. Since, we try to provide efficient VCR playback for the users we select Linear Network coder over Random Network coder.

There were very few works done in media streaming with Network Coding in the past, but recently, we find that Network Coding in media streaming has gained momentum due to the fact that it is robust and theoretically claimed proof benefits have provided ample opportunity in this field (Wang and Li, 2007). Hence, we study its impact in media streaming with VCR functionality that enhances performance through a proxy by generating packets from network coded packets as the same way as error packets are generated. This has not been dealt so far. We try to build on the available VoD systems that are present for working with NC.

Related works: Peer-to-Peer file downloading and streaming have become popular internet applications. In this project, characteristics of a video streaming system in a Peer-to-Peer environment are understood and an efficient system that incorporates those concepts has been implemented (Wang and Ansari, 2011).

The major components that are present in mostly all Peer-to-Peer streaming systems are: (1) a set of servers as the source of content (e.g., movies); (2) a set of trackers to help peers connect to other peers to share the same content; (3) a bootstrap server to help peers to find a suitable tracker (e.g. based on which geographical region the peer is located) and to perform other bootstrapping functions; (4) other servers such as log servers for logging significant events for data measurement, as mentioned in (Li *et al.*, 2012).

A number of P2P live streaming systems are deployed with high viewing quality but low server burden, including Cool Streaming, PPLive, PPStream, UUSee, AnySee and Joost etc (Liu *et al.*, 2010). However, there are vital differences between live streaming and VoD streaming (Li and Niu, 2011). For example, users' interactive behaviours like pausing and random jumping are allowed when they are subscribed to VoD services but live streaming systems do not provide these features. As a result, the design and deployment of a real world P2P-VoD system is more difficult than a P2P live streaming system. In fact, on demand video streaming is not a new topic and the research begins since early 90's. IP multicast based proposals like patching, periodic broadcasting and merging faced the deployment problems of IP multicast. Later on we find that, there are a number of proposals for peer assisted Video-on-Demand streaming

such as tree-based approach and overlay tree based patching were also developed.

The ability to deliver large amounts of data at a reduced deployment cost is one of the reasons why P2P systems have become quite popular (Papadimitriou and Steiglitz, 1998). The inherent self organization and resource scalability available in such environments when utilized can prove really beneficial. Moreover, our system working with the proxy server provides improved performance than the Dynamic Skip List Based (DSL) system proposed by Wang and Liu, 2008. This work of ours manages heterogeneous peer systems which have not been dealt in DSL.

Loss of media packets is a common problem and this could be attributed to events of network congestion and transmission over unreliable channels (Wang and Li, 2007). The media content when transmitted properly makes sure that the reconstructed video becomes less susceptible to error propagation.

Network Coding was first proposed in 2000, in the information theory community. From 2005, it is a research topic in Peer-to-Peer networks. In the works of (Li *et al.*, 2012), it has been analysed that Network Coding is more suitable for Peer-to-Peer steaming than in Peer-to-Peer sharing. Network Coding in multicast streaming has improved the throughput (Chan *et al.*, 2010; Gkantsidis and Rodriguez, 2005; Huang and Zhang, 2011).

Based on these understandings, we find that (1) to increase playback time, the network conditions should be optimal and transmission losses should be dealt with and (2) for tackling the losses, we incorporate NC (Chan *et al.*, 2010), which in turn increases the processing at the client side, causing delay in playback time.

MATERIALS AND METHODS

The study of Network Coding in P2VoD has led to the setup of a Peer-to-Peer (P2P) network with video streaming. The basic system set up is that of a Peer-to-Peer environment with Network Coding. The initial repository of all the video files is present in a Media Server that acts as the video streaming server as well for the P2VoD. The media server is then connected to the P2P network through proxy server as shown in Fig. 1. The proxy server has a list of all the peers and the video files present in each of them. This way if a peer1 requests for a video which is already with peer2, then the request is directed to peer2, instead of wasting the server bandwidth. This way, only peers requesting for new videos are allowed to connect to the server and the later requesting peers share content among them.

The detailed architecture of the video streaming system in which the Network Coding techniques are applied is explained in this section and it is as shown in Fig. 2.

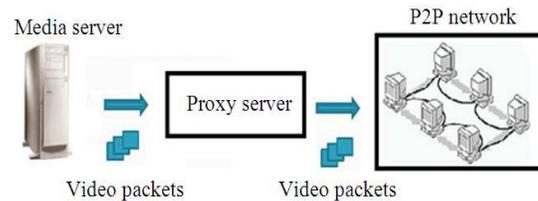


Fig. 1: System design

Media server: In any VOD system, there is a Media Server. In our system too, available movies (video files) are stored in the media server and based on the peer request; the video is retrieved and transmitted to the peer. The server has a user interface. It allows the administrator to upload video files on to the server. The video files are split into packets for transmission.

Request handler: The peer request is received at the request handler unit. The request handler gets the requestor peer name, requestor peer IP and requested video file name. The file name is then sent to the video storage unit.

Retrieval/storage of video files: This unit helps to upload the video files into the server. The files are all stored in the video storage of the media server. When a file name is received from a peer, then the retrieval unit sends the request to the video storage to find if the file is available in the media server. If the file is found, then the file is split into packets and the packets are sent to the request handler unit. The request handler unit then directs the packets to the requesting peer.

Algorithm:

```

Input: Requested video by the user
Output: Identified video frames sent by server
Begin
Search for video in the database
    If not found return null
    Else
        Check if first frame or nth frame
        Transmit the requested frame after splitting video
Begin
Sz = size of the video
max = maximum packet size allowed
Read file as byte array
Len = Sz / max
Save the video from (part-1)*Len (number of bytes)
End
End
    
```

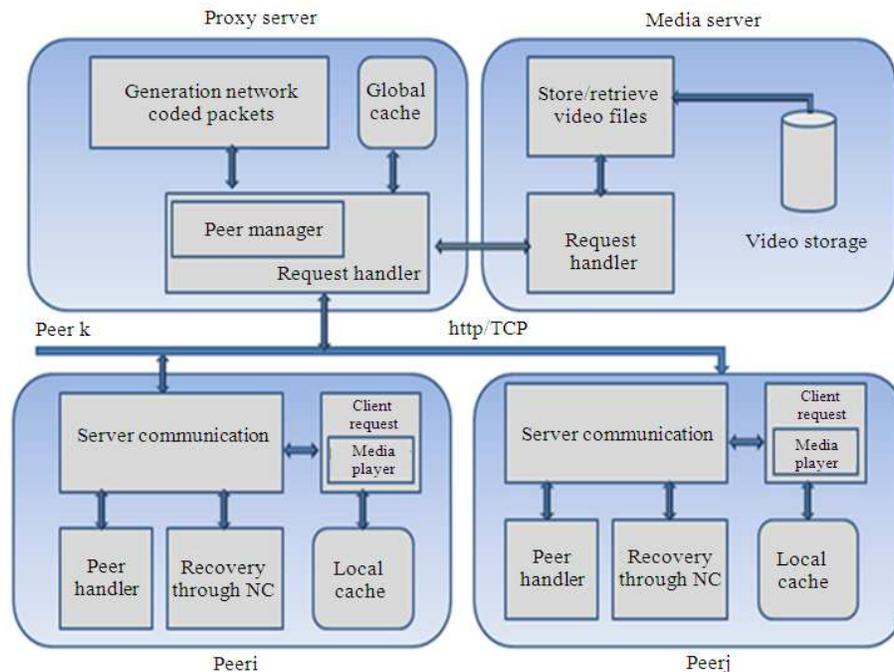


Fig. 2: Architecture of video streaming system

Video storage: The video files that are uploaded into the media server are stored in the uploaded videos folder in the server. When the file is present into the storage folder then the file is forwarded to the store/retrieve of video files for the packet splitting. If the file is not present, a message is sent from the request handler to the peer, to communicate to the user that the file is not uploaded into the unit. In this way the video storage clarifies with the uploading peer.

Proxy server: This acts as a mediator between the server and the peer clients. Peers send their requests to the proxy; the proxy in turn sends the requests to the media server. Filtering mechanisms such as firewall are placed at the proxy based on the system requirements. This will restrict the requests that are sent to the client. The different components in our proxy are explained here. The main task of the Proxy is to update and manage the global cache, where it stores information of the requesting peers and their requested content and the uploading peers and their contents. It maps the peers to their requested content placed among them to avoid transmission from the media server. The Network Coding generation is done in the proxy server.

Request handler/peer manager: The proxy server gets the request from peer and checks its global cache to see if any other peer is already having the requested video. If the video is already available with a peer, then the

request is transferred to that peer by the peer manager request module. The peer acts as the gateway between the peers and the server. It controls which peer communication to the server by only letting peers which request for new videos to talk to the server.

Generation of network coded packets: In the current scenario, a network coder block has been placed in the proxy. The network coded packets are transmitted to the requestor peer along with the original packets. If any of the packets are found missing from the original message in the receiver end, then we use the network coded packets to reconstruct the lost packets.

We use Linear Network Coding (Chan et al., 2010) here, where intermediate nodes are used to send message from source to destination. The Linear Network Coding method of Hamming Code is used here. In Hamming Code, the processing of encoding the message is as follows. The video is converted to its binary format. The binary data is taken and it is padded with some extra parity bits. A parity bit is placed at every $2k^{\text{th}}$ location (i.e., 1, 2, 4, 8, etc., the value of k ranges from 0 to length of file- n). The value of k is increased from 0 to the n . This encoded data is transmitted.

At the receiver side, the parity is once again verified with every $2k$ bit as shown in Fig. 3. If any bit is corrupted during transmission, then it can be found and corrected. The data is then converted back to video peer networks working is explained in the following sections.

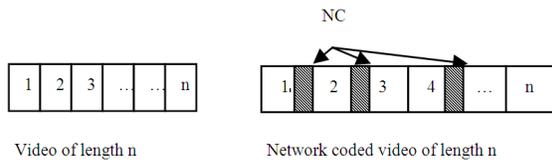


Fig. 3: Network Coding Packets Piggybacked

Client request/media player: Every single peer instance has a user interface which allows the user to request for a particular video file. The details of a peer such as: peer name, peer IP address and the requested file name are sent to the server communication unit. When the peer receives the video packets in response to a request, then the video packets are played in the media player, which has been designed using Java FX software.

Server communication: The details of the peer, peer name, peer IP address and the requested file name are taken and passed on to the proxy server, by the communication module. This module handles the request that are issued to the peer and also the ones that go out of it.

Peer handler: When peer1 requests the proxy for video1 which is already with peer2, then the request is passed to peer2. Now the peer2's peer handler unit is used to transmit the requested video packets to peer1.

Local cache: The video files that get delivered to the peer are stored in the local cache. When the user wishes to play the video, then the video packets from this folder are taken and played in the media player.

Network coding module: When a peer1 requests for a video available at peer2, then the proxy transfers the request to peer2. Now peer2 acts as the server and sends the video in packets form to peer1. Before sending from one peer to another, the video undergoes the Linear Network Coding process. The network coded packets are sending to the requestor peer along with the original message. During the transmission between peers, if there is any packet loss or some packets get corrupt, then the packets can be regenerated by using the network coded packets.

Experimental setup: Network Coding, used in Peer-to-Peer networks, is not only for better information flow, but also for error correction. In the media streaming scenario, a Network Coding block has been placed in each peer. The network coded packets are transmitted to the requestor peer along with the original packets. If any of the packets are found missing from the original message in the receiver end, then we use the network coded packets to reconstruct the lost packets.

Initially a network with a fixed set of nodes is created. We use Linear Network Coding where a few intermediate nodes are used to send message from source to destination. The Linear Network Coding method of Hamming code is used here. In Hamming code, the processing of encoding the message is as follows. The video is converted to its binary format. The binary data is taken and it is padding with some extra parity bits. A parity bit is placed at every 2^k location. The value of k is increased from 0 to the size of the file. This encoded data is transmitted.

At the receiver side, the parity is once again verified with every $2k$ bit. If any bit is corrupted during transmission, then it can be found and corrected. The data is then converted back to video. The pseudo code for performing Network Coding of the video packets at the different nodes can be explained as follows: Let, P_i be the i^{th} peer, Number of packets received by P_i are $1 \dots k \dots n$, viz., n packets. Let, m , be the number of parity bits in the k^{th} packet, the number of bits in packet are $2^m - m - 1$. So the numbers of bits that get transmitted are $2^m - 1$.

Algorithm:

Input: k^{th} packet
 Output: Encoded k^{th} packet
 Begin
 At the transmitter end:

- Generate the data array (binary data) from the video packet
- Insert check bits at every power of 2 location (i.e., 1, 2, 4, 8, etc., the value of k ranges from 0 to Len (length of file (n)))
- Parity of bits between two check bits is calculated and saved at the 2^k location
- This data array is transmitted

At the receiver end:

- Parity for every power of two (1, 2, 4, 8, etc.) is calculated
- If any bits don't match, their positions are recorded (w_i)
- Sum the positions of the w_i bits is calculated, as $W = \sum \text{position}(w_i)$
- The sum is the position of the incorrect bit
- Flip the value of the bit in that position

The evaluation parameters that have been considered for the evaluation of the system are response time, bit rate and packet loss recovery for normal video playback as well as for VCR playback.

Table 1: Evaluation parameters

Simulation parameters	Values
No. of peer nodes	10
No. of movies	5
Movie length	47-65 seconds
Packet length	10-5 seconds
Bandwidth	256-2000 kbps
Simulation time	1 hour

We implement our system with the available resources as shown in Table 1. The implementation done on these available peers has been studied which would give more insight on the working of network coded media streams.

RESULTS

The study with limited resources as presented in Table 1 does not make the system invalid, because here we look in for the changes that occur within limited peer groups and that are evaluated with comparison from non network coded streams to network coded streams in linear coded pattern. This, highlights the major part of our work which brings about the need for network coded streams are required not only for huge peer networks but also for the smaller peer networks. This provision is not dealt with by most of the researchers trying to achieve benefits with NC.

The different systems with which the evaluation was done are:

- Normal P2P Network
- P2P Network + NC(with Base64 video encoding)
- P2P Network + NC(with Linear Coding)

For measuring the response time, the system run was done in all the different situations by increasing the number of peers. The response of the different systems is as shown from the graph in Fig. 4.

DISCUSSION

In Table 2, during the testing for the module with Network Coding, one packet was removed from the received video. The peer which received this data realized this missing packet and decodes the NC packets for that missing packet alone. Hence, response time of the module with NC function increases, as all the other processing such as detecting the missing packet and then decoding them consumed more time. The above figure shows the working of the system for normal playback. The behaviour of the system while operating upon different VCR operations at static intervals simultaneously by all peers has been tabulated.

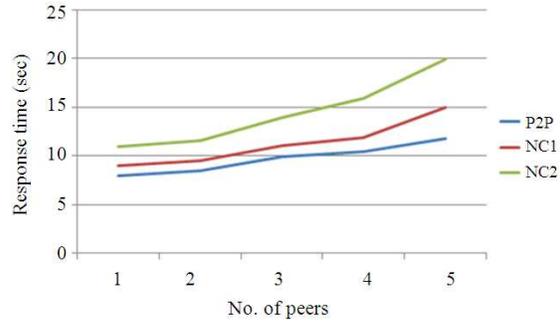


Fig. 4: Graph shows response time VS number of peers during normal play back operations

Table 2, has the values captured when requests for video was sent out from peer1 to peer10 simultaneously. Peers that were delivered corrupt packets and had to perform Network Coding to extract the original packets are Peer9 and Per10 in column-1, Peer6 in row-4, Peer5 in row-9 and Peer4 and Peer10 in row10. The values tabulated at these cells for the respective peer show time stamped values that are little bit high in time lapse than the others those have almost equal distribution.

When video requests were sent out at random time intervals which we mention here as dynamic VCR request time, the response of the system was in similar lines of the Static (regular time interval) response. In Table 3, Peer5 in row-2, Peer8 in row-5, Peer9 in row-8 show the response times when the system performs Network Coding. From the values it becomes clear that the delay in transmission between the packets that have been regenerated through NC and the ones that have been received unaltered is 0.4% to 2%. This indicates that the NC streaming of video packets is better for even smaller peer network groups. It is highly possible if NC is incorporated by commercial structures to handle lower infrastructure groups they could benefit with this system. Further, Table 4 shows, the various timestamps obtained for VCR operations with NC this is more benefit from the user's perspective.

From the graph shown in Fig. 5, we can come to a conclusion that, the response time of the peer requests becomes more when the number of requester peers are increased. Also when the amount of coding increases, in the following ascending order, P2P system, P2P with Network Coding, we can notice that the response time also increases as the processing time increases.

Without Network Coding, we can understand that the packet loss recovery is nil. Network Coding helps in effective playing of video despite some missing packets.

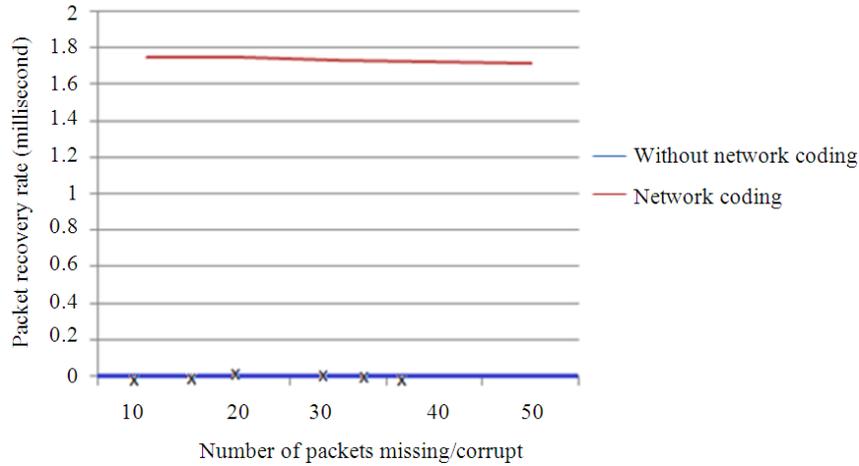


Fig. 5: Graph shows the system ability with normal playback with and without NC

Table 2: VCR request time (static) and playback time at different peers

VCR Request sent out time-static (ms)	Playback at									
	Peer1	Peer2	Peer3	Peer4	Peer5	Peer6	Peer7	Peer8	Peer9	Peer10
1	1.2	1.10	1.40	1.20	1.25	1.50	1.12	1.80	1.70	1.80
5	5.3	5.33	5.50	5.45	5.70	5.55	5.80	6.00	6.20	6.30
10	10.1	10.10	10.40	10.50	10.70	10.40	10.55	10.80	10.10	10.50
15	15.5	15.10	15.22	15.23	15.25	15.10	15.01	15.25	15.01	15.15
20	20.5	20.50	20.45	20.20	20.20	21.40	20.55	20.75	20.50	21.20
25	25.2	25.40	25.80	25.90	25.30	25.40	25.20	25.75	25.50	26.11
30	30.8	30.90	30.45	30.25	30.70	30.55	30.80	31.00	30.80	30.20
35	35.3	35.10	35.45	35.25	35.50	35.15	35.12	35.70	35.80	35.30
40	40.2	40.40	40.22	40.45	41.10	40.85	41.00	40.40	41.20	40.10
45	46.8	45.60	47.50	48.50	45.70	47.40	45.55	46.75	46.50	47.20

Table 3: VCR request time (dynamic) and playback time at different peers

VCR Request sent out time-dynamic (ms)	Playback at									
	Peer1	Peer2	Peer3	Peer4	Peer5	Peer6	Peer7	Peer8	Peer9	Peer10
10	11.00	10.50	10.80	10.90	11.30	11.40	10.55	10.75	10.5	11.20
17	17.50	17.50	17.45	17.20	17.20	18.40	17.55	17.75	17.50	18.20
18	19.20	19.10	19.40	19.20	19.25	19.50	19.12	19.80	19.70	19.80
23	23.20	23.40	23.22	23.45	24.10	23.85	24.00	24.40	24.20	24.10
31	31.80	31.90	31.45	31.25	31.70	31.55	31.80	32.00	31.80	32.30
43	43.30	43.33	43.50	43.45	43.70	43.55	43.80	44.00	44.20	44.30
58	58.10	58.10	58.40	58.50	58.70	58.40	58.55	58.80	4.10	4.50
61	61.50	61.10	61.22	61.23	61.25	61.10	61.01	61.25	65.34	64.30
70	70.20	70.40	70.80	70.90	70.30	70.40	71.20	70.75	70.50	71.11
82	82.30	82.10	82.45	82.25	82.50	82.15	82.12	82.70	82.80	82.30

Table 4: Response times for various VCR operations

Fast forward request at time	Response time in secs	Backward seek Requests at time	Response time in secs	Backward seek Request at time	Response time in secs
5 sec	2.71	5 sec	1.92	5 sec	3.50
1 min	3.21	1 min	2.87	1 min	4.67
5 mins	6.17	5 mins	5.01	5 mins	5.25
10 mins	9.50	10 mins	6.35	10 mins	8.10
15 mins	11.51	15 mins	7.00	15 mins	4.50

CONCLUSION

In this study, we study how linear Network Coding can be used in Peer-to-Peer networks to improve video efficiency while performing VCR operations. The system is able to provide error-free video data to the end user in spite of packet loss or corrupt packets instantaneously. We conclude that Network Coding can be implemented in P2VoD networks to enhance transmission.

Advantages:

- No need for retransmission of the lost packets. Time is saved, as response time is less, when compared to systems that ask for retransmission if there are any missing packets.
- When the original packets of the video are lost in the network, the packets can be retrieved from the network coded packets that are present in the peers.
- When the user is performing VCR operations like Fast-forward, Backward Seek and Random Seek, if any packets are corrupt, then efficiency would be lost, but having near instantaneous recovery with network coding helps in a dynamic network, where there are flash crowds or peer departures.
- As the need of retransmission is not necessary, the network congestion is eased in larger (>10000 peers) networks.

Disadvantages:

- Wherein, piggybacked NC packets incur transmission delay
- Need larger cache in for holding NC packets from different streams
- Initial playback latency is quite dominant in all NC incorporated systems, hence the time delay for initial playback is seen to be more than normal video streaming

In future, this system can be further enhanced by incorporating the randomised network coding and study its various patterns with respect to VCR operations. This can be extended to large P2P systems and review their effect on NC.

REFERENCES

Chan K.H.K., S.H.G. Chan and A.C. Begen, 2010. SPANC: Optimizing scheduling delay for peer-to-peer live streaming. *IEEE Trans. Multimedia*, 12: 743-753. DOI: 10.1109/TMM.2010.2053524

Gkantsidis, C. and P.R. Rodriguez, 2005. Network coding for large scale content distribution. *INFOCOM 2005. Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Mar. 13-17, IEEE Xplore Press, pp: 2235-2245. DOI: 10.1109/INFCOM.2005.1498511

Ho, T., M. Medard, J. Shi, M. Effros and D. Karger, 2003. On randomized network coding.

Huang, J. and X. Zhang, 2011. Performance comparison between P2P VoD with and without network coding. *Adv. Comput. Commun., Control Automation*, 121: 641-648. DOI: 10.1007/978-3-642-25541-0_81

Li, B. and D. Niu, 2011. Random Network coding in peer-to-peer networks: From theory to practice. *IEEE J. Magazines*, 99: 513-523.

Li, B., Y. Feng and B. Li, 2012. Rise and fall of the peer-to-peer empire. *Tsinghua Sci. Technol.*, 17: 1-16.

Li, S.Y.R., R.W. Yeung and N. Cai, 2003. Linear Network Coding. *IEEE Trans. Inf. Theory*, 49: 371-381. DOI: 10.1109/TIT.2002.807285

Liu, Z., C. Wu, B. Li and S. Zhao, 2010. UUSee: Large-scale operational on-demand streaming with random network coding. *Proceedings of the IEEE*, Mar. 15-19, UUSee Inc. San Diego, USA.

Nguyen, K., T. Nguyen and S.C. Cheung, 2007. Peer-to-peer streaming with hierarchical network coding. *Proceedings of the IEEE International Conference on Multimedia and Expo*, Jul. 2-5. IEEE Xplore Press, Beijing, pp: 396-399. DOI: 10.1109/ICME.2007.4284670

Niu, D. and B. Li, 2007. On the Resilience-Complexity Tradeoff of Network Coding in Dynamic P2P Networks. *Proceedings of the 15th IEEE International Workshop on Quality of Service*, Jun. 21-22, IEEE Xplore Press, Evanston, pp: 38-46. DOI: 10.1109/IWQOS.2007.376546

Papadimitriou, C.H. and K. Steiglitz, 1998. *Combinatorial Optimization: Algorithms and Complexity*. 1st Edn., Dover Publications, Mineola, ISBN-10: 0486402584, pp: 528.

Thomos, N. and P. Frossard, 2009a. Network coding and media streaming. *J. Communi.*, 4: 628-639. DOI: 10.4304/jcm.4.9.628-639

Thomos, N. and P. Frossard, 2009b. Network coding: from theory to media streaming. *J. Commn, Spl. Issue Multimedia Commn., Net., Appli.*, 4: 628-639.

Wang, D. and J. Liu, 2008. A dynamic skip list-based overlay for on-demand media streaming with VCR interactions. *IEEE Trans. Parallel Dis. Sys.*, 19: 1-12.

- Wang, M. and Li B, 2007. Network Coding in live peer-to-peer streaming. *IEEE Trans. Multimedia*, 9: 1554-1567.
- Wang, N. and N. Ansari, 2011. Downloader-initiated random linear network coding for peer-to-peer file sharing. *IEEE Syst. J.*, 5: 61-69. DOI: 10.1109/JSYST.2010.2063377
- Yu, X., J.W. Modestino and D. Fan, 2007. Joint multiple description coding and FEC for delay-constrained applications in congested networks. *Proceedings of the IEEE Information Theory Workshop*, Sept. 2-6, IEEE Xplore Press, Tahoe, pp: 547-552. DOI: 10.1109/ITW.2007.4313133