International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 4, Issue 4, July - August 2015 ISSN 2278-6856

Performance Analysis of Buffer management for live streaming in P2P VOD systems for mesh based non-cluster and cluster environment

Dr. Nemi Chand Barwar[1] , Dr. Rajesh Bhadada[2]

[1]Associate Professor Deptt. of CSE, Faculty of Engineering, J N V University,Jodhpur

[2] Associate Professor Deptt. of ECE, Faculty of Engineering, J N V University,Jodhpur

Abstract

This paper explores mesh based clustering for different start video streaming in P2P systems and estimates the performance of non cluster and clustered models. These models are based on mesh based topology of P2P streaming consisting of peer join/leave. A new approach by way of "clustering" peers has been proposed to tackle P2P VOD streaming. The proposed models were simulated and verified using OMNET++ V.4. A mathematical model has been formulated to evaluate the performance metrics such as buffer size, buffer exchange period over frame loss ratio, startup delay and end to end delay. The results show that(i) unpunctual viewers can be enabled view missing part of video from beginning during live streaming, (ii)the associated buffer in conjunction with join/release time provides constant buffer size and thus ability to provide better service to viewers,(iii)Starting and play out delay gets bounded as the number of peers predefined,(iv) clustering minimizes startup delay, end to end delay and frame loss as compared to existing systems, which are important parameters of a P2P system for streaming.

Keywords: Peer-to-Peer (P2P), Video streaming, Video on Demand(VOD).

1. INTRODUCTION

Conventional Client-Server model has been followed for video streaming over internet, resulting in many limitations in terms of the capacity of servers and bandwidth provided by them. It is not possible to add or contribute by clients even if they have spare resources. Such systems are based over centralized servers and prone to single point failure. Further resources and their utilization available under internet could be highly unbalanced to be able to provide sustained stable service to clients with limited bandwidth [11].

The P2P content distribution systems (like BitTorrent[2]) and Voice-over-IP systems (like Skype[6]), has enthused for application of P2P networking technique for streaming of video data. In P2P system, a peer not only downloads data from the server and other peers, but also uploads the downloaded data to those who require it. This brings in lot of bandwidth saving and reduce burden on the original servers while simultaneously generate peers with acceptable service.

Depending upon whether video is generated live or used from pre-stored, video streaming on P2P networks is termed as P2P live streaming and P2P VOD respectively. The former provides certain live video programs and the latter enables flexibility on choosing any programs and any parts of the programs.

Consistent enhancements at network bandwidths have made emerge IPTV[6],[7]. In contrast IP multicast P2P networks implement multicast simply in end systems above the application level. Many models have been tried to achieve high viewing quality, lower server burden, scalability and reliability. Mile stone P2P live streaming systems[1],[5],[6],[8],[9],[10] are Cool streaming, PPLive, PPStream, UUsee, Anysee and Joost etc.

P2P VOD is an emerging service to let users select a video or different part of an available video. P2P-VOD can bring substantial reduction in server loading. Many P2P-VOD versions have been tried such as PPLive,Joost, GridCast, PPStream, UUSee etc. however their architectures has broadly been grouped either as tree-based or mesh-based structures.

2. BACKGROUND

Architecture of P2P VOD streaming can be classified either of two types: tree or mesh structured.

(A) Tree-Based Systems: In this type, peers build up a tree structure at application layer and original server is placed at root of this tree. Other clients in the network have only one parent and may have several children. Streaming server at root encodes new packets and send them to all its children, while similarly each node gets packets from the parent and forwards them to their children. This mechanism is a push approach and here no requests and control packets are sent to the parent nodes. Tree depth decides delay due to number of hops a packet has to cross. Tree-based system lacks strength that building up optimal tree takes more time and needs more knowledge of available network resources including bandwidth. Further dynamic changes in network makes task of maintenance complex and hence the reorganization process needs more time. Moreover, when clients offer small buffers or nodes move from the top of the tree (low playback delay) to the bottom (long playback delay), it may generate interruption at video playbacks.

(B) Mesh-Based Systems: Under heterogeneous networks, clients keep joining and leaving the network frequently, needing the tree to be repaired or reorganized very often. Under such condition, Mesh-based systems can better handle the networks and the unstable user behavior. Here a client keeps searching for possible sources in the network to download needed video segments. To choose right peers, a client has to know where the needed segments are stored. It can be done by the exchange of information between the clients called buffer maps (bit vectors which indicate the availability of segments). With help of bit vectors, one of the sources is selected and client can request the peer to send this segment. To organize these exchanges control traffic is generated as overhead. Generally the clients have option to download the same segment out of many peers, making Mesh-based systems more resistant against node failures.

H. Ketmaneechairat et al[9] have proposed a non clustered mesh based model for broadcasting in P2P system consisting of five stages. We have extended this model for video streaming by way of non clustering and clustered model for analyzing performance of P2P Systems.

3.Proposed Clustering

The strategy of proposed cluster-based system architecture and design is described under following sections. Whenever a new peer join and download chunks from the peers under this model for different start video, a global tracker decides that which cluster and node will be best to join.

3.1System Architecture

In this approach, peers (nodes) shall be grouped according to their joining time or the chunks available with them. It will have a server, a global tracker (GT), super node (SN) or local tracker (LT), backup-node (BN) and normal nodes (NN, seed and leech). The server shall be a special node that provides all chunks of a live video for streaming. The global tracker maintains the list of all super nodes and known to all nodes. A super node maintains list of all nodes in a cluster and functions as a local tracker. All such super nodes are connected with the global tracker to perform function of synchronizing the lists of all nodes in the cluster. The SN, NN and BN will used for performance of downloading (leech) and uploading (seed) of chunks.

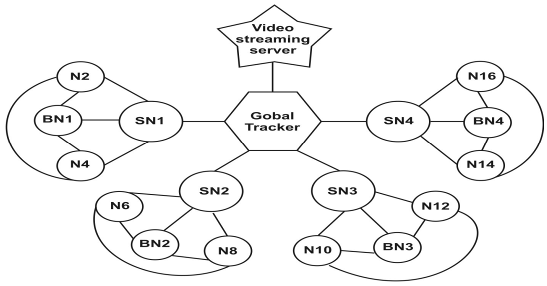

Figure 1: Proposed clustering of nodes for video streaming

3.2 Cluster-Based System Design

In this model the clustering implies node grouping in terms to their network proximity, where this proximity is determined by the joining time of each node and the availability of chunks. In this model, it has been assumed that there is no shortage of bandwidth at peer for upload/downloading of chunks and many users joins within a short interval.

Based on the non-cluster model for P2P video streaming[8],[9], the methods of peer exchange information, peer selection, buffer organization, segment scheduling are not changed but extended by a logical clustering mechanism. The clustering is realized by the separation of nodes, super nodes, backup-nodes and the introduction of local trackers.

The proposed model presumes to have logically a global section and local clusters. The global section comprise of (i) the original server (source of the video for stream) and (ii) a global tracker (known to all nodes in network). Each cluster formed will be comprised of a super node (SN), a local tracker (LT), a backup-node (BN) and normal nodes (NN). SN also does the task of the local tracker i.e. it acts as coordinator amongst nodes in respective cluster. The super node is the only node in the cluster who shall download from the server and other super nodes (of other clusters). The LT provides and maintain neighbor list within cluster. NNs within cluster can exchange chunks with own cluster nodes only and they are not known outside their cluster. Whenever SN leaves cluster, BN becomes the new super node. The BN also has to update and maintains list of all nodes in the cluster. Any NN can become the SN or BN. Selection of a node as SN or BN is explained under the next sections. This proposed cluster based system model will offer two major benefits-(i) the clusters being merely logical entities under control of different known neighbor lists and the resulting chunk exchange paths and (ii) it bring in 3 levels of chunk sources (server, super nodes, nodes) that enable a control over distribution of traffic and thus results in reduction in load of server and tracker (they are dedicated parts in system). Figure 1 exhibits cluster-based system model.

3.3 Main System Processes

Cluster based system need five main processes to make system function as under-

(1) Process for Peer Join

To ensure a new node join a video stream and participate with neighbor peers (in the cluster), it has to first contact the GT for allotment of a cluster and its SN. As soon the new node approach GT with request to receive the first chunk, the GT contacts SN of each cluster to locate a node(s) in possession of the first available chunk. GT The cluster which offers maximum number of nodes containing the first chunk is selected by GT for allotment and provides the address of the related LT of SN to enable new node to register there.

Thereafter the new node contacts the allotted SN's LT, then LT returns a random list of neighbor peers in the same cluster. Now the new node sends the message to these neighboring peers to exchange buffer maps and finally select neighbor peer to download chunks.

(2) Process for SN Selection

Initially as and when the first node joins, the GT will make first node to be SN+LT of the first cluster, and limits the cluster size to *m* nodes. Whenever GT gets a joining message from a new node, it checks the cluster size to select appropriate cluster and also verify the member count of the cluster chosen, and if it is less than *m*, the address of SN of chosen cluster is sent to the joining node. In case count of the selected cluster is full (=*m*), a new cluster may be created by splitting a node that has first chunk available in the old cluster to be a SN for a new cluster by GT. If there is no cluster in the system, the first cluster is created and the first joining node will be a SN of the first cluster.

(3) Process for BN Selection

Any NN can be chosen as BN. Selected BN shall also keep a list of all peers in the cluster by frequent contact with SN. Whenever the SN leaves the cluster, the BN becomes the new SN. The BN out of NN can be chosen by any of three criteria as under-

(a) Choose the node who has joined the cluster next after the first node (the 2^{nd} joining node) OR
(b) Choose the node who has joined at the middle in the group (the $\frac{m}{2}$ th node) OR
(c) Choose the node that joined last in the group (the n^{th} node)

The first criteria offers equal chance to all nodes to be a SN and BN, along with the disadvantage of frequent SN and BN selection. In comparison the second criteria results in not so often selection of SN and BN. The third criteria brings a new super node least often but may cause packet losses. The second criteria has been assumed for implementation in the simulation.

(4) Download Process

On joining of a new node in the cluster and its contact with the SN, the LT of SN passes a random list of peers from the same cluster and the new node can only download from these neighbor peers. The buffer at each node shall be having three sections: data buffer, buffer map and sliding window. The data buffer holds video frames, while the buffer map is bit vector representation of the information of available segments on a node. Periodically all partner nodes exchange their buffer map with other partners. The peer decides which partner nodes to be used to get needed video chunks on the basis of buffer maps. In case needed chunk being available with more than one partners the peer will select the partners with minimum delay or maximum bandwidth. The sliding window is used for storing a number of displaying segments to maintain display continuity, while the starting delay of each node will be bounded. In the proposed model, a circular buffer has been assumed for buffer management.

(5) Process for Node Leaving

When a node leaves the cluster, the LT removes it from the list of peers. In case the leaving node happen to be a LT (and SN), then BN is updated as LT (and SN). Whenever the last node leaves the cluster, the cluster is deleted. The LT essentially updates the GT about leaving nodes to synchronize the list of SN there. This process tackles three types of nodes leaving differently as per leaving node being SN, BN or NN. When SN exits the cluster, it sends flooding message to all nodes. All nodes in the cluster respond and send keep-alive message to their SN and BN. The SN sends keep-alive messages to the GT. When BN exits, it sends the message to inform the super node. In case of exits by NN, it can be done any time without process.

4. MATHEMATICAL MODELLING

In P2P environment of proposed model under investigation, there exist four types of delay: (i) startup delay[T_{STUPD}]: the time that the node has to wait for arrival of first frame in the buffer but not being displayed (ii) Playback delay[T_{PLAYBK}]: the time that elapsed, which node waited to fill chunk in the buffer for smooth playback (iii) Joining delay[T_{JOIN}]: the time that the node spent to join late in the streaming depending upon server time (iv) Starting delay[$T_{STRGDLY}$]: the total elapsed time in wait that user spends to show the first frame.Under this analysis, the starting delay and buffer size has been estimated as under-

4.1 Starting Delay Estimation

The T_{STUPD} depends upon the number of neighbors used for locating video selection for download. The two cases (i) only one available neighbor (ii) more than one neighbors have been considered for startup. The startup delay is directly dependent upon the network parameters i.e. bandwidth and delay [effect of processing delay and queuing delay has been neglected in calculations].

(a) Only one neighbor case-

T_{STUPD} = Propagation delay + Delay due to transmission

$$T = \sum_{i=1}^{n} Propagation\ Delay + \sum_{i=1}^{n} \frac{Packet\ Size}{Transmission\ Rate_i} \qquad (1)$$

Where *n* is the number of links from the sender to the receiver.

(b) More than one neighbors case-

From Eq. (1) Delay = T_{STUPD}

As, there are many neighbors, there may exist many paths. The neighbor that has the smallest delay will deliver first frame. Here p_i denotes symbolic path from the i^{th} neighbor. The time that is needed to download the first chunk depends on transmission delay and the propagation delay necessary for the sender to provide required packet, *m* is the total number of neighbors.

$$T_{STUPD} = \text{Minimum amongst } (Delay_{p1}, Delay_{p2},\ldots,Delay_{pm}) \text{ or} \quad (2)$$
$$= \text{Min } (Delay_{pi}) : i = 1,2,3,\ldots,m$$

Starting Delay ($T_{STRGDLY}$): It depends upon join time, playback delay and startup delay. The starting delay is estimated thru Eq. (3).

$$T_{STRGDLY} = \text{Max } (T_{JOIN}, T_{PLAYBK}) + T_{STUPD} \quad (3)$$

4.2 Buffer Size Estimation

An estimation of buffer size is needed at time of joining while also when releasing. While joining, the buffer size at each node will be different where while release, the buffer will have the same size at each node. It is more practical to measure size of buffer in time (seconds) in VOD systems as the wait time to display of video depends on the fill rate from peers.

The video stream viewing process depends upon joining delay. Consequently a peer joining late may not be able to view the whole stream. Such fraction of video stream length time is used to determine maximum threshold for the joining delay as QoS indicator. When a user is able to join before the fraction of video stream length time value, it is under mixed mode i.e. live-video streaming as well as video-on-demand. However when a user is able to join after the fraction time, such user will be able to view live-video streaming only. A threshold value (*h*) as % of total display time is useful for this fraction time as it may be varied as system parameter to allow according to latest joining i.e. control parameter for system still in hybrid (live + VOD) mode.

The maximum number of chunks that receiver is able to receive in a unit time, is available buffer.

$$\text{I: Buffer Size} = \text{MAX } (T_{JOIN}, T_{PLAYBK}) + \text{Available Buffer} \quad (4)$$
$$\text{II: Buffer Size} = \text{MAX } (T_{RELEASE}, T_{PLAYBK}) + \text{Available Buffer} \quad (5)$$
$$\text{Available Buffer} = \text{Link Speed (bandwidth)/Fill Rate} \quad (6)$$

The release buffer denoted $T_{RELEASE}$, is the time to wait until buffer is released.

$$\text{Buffer Size} = T_{PLAYBK} + \text{Available Buffer} \quad (7)$$

The buffer size estimation is calculated as following-

(a) In case when the fraction of video stream length time is less than *h*, than buffer size will be computed as per in Eq. (4) and Eq. (5)
(b) In case when the fraction of video stream length time is more than *h*, than buffer size will be computed as per Eq. (7).

Since, the available buffer is used to receive new frame, the playback buffer and release buffer are constant. The fill rate refers to the number of chunks send to buffer within one second.

5. SIMULATION SETUP

To investigate the proposed clustered model, a setup was prepared to conduct simulations under clustered and non clustered scenarios using discrete event network simulator, OMNET++ version 4.1. INET frame work is a collection of network units containing models for internet protocols such as IP, TCP, UDP and data link layer which work in OMNET++. 'Oversim' is an open source frame work for implementation of P2P network in OMNET++, containing models for unstructured network (mesh network). A program in OMNET++ configured mesh based P2P media streaming as non-clustered and clustered model for this work.

5.1Simulation Configuration

Simulations have been configured with network sizes (varying between 25 to 500 nodes), peer churn and number of neighbors. For simulating valid video stream 'Star Wars IV' trace file was applied. Under different clustered and non-clustered configurations, simulations were programmed to run 5 times and average of all 5 run was taken as the peer's output for each scenario.

The simulation physical topology is generated using Georgia Tech Internet Topology Model (GT-ITM) tools for OMNET++ V.4 with 28 AS(backbone routers) and 28 access routers per AS in top down mode.

A peer selects a router randomly and connects to it by a random physical link. Each peer selects number of neighbors as per the configuration of experiment. Neighbors exchange buffer maps, each window of interest includes segments, each one of 1 second. Video file is divided into chunks. In this simulation it has been assumed that size of each chunk is equal to one second of player length.

5.2 Measured Parameters

To investigate performance of proposed model of P2P video streaming for mesh type non cluster model as well as clustered model have been carried out via simulations. Following network performance parameters were measured and analyzed.

1. **Frame loss ratio:** The ratio of the dropped frames with respect to total video frames transmission (% of frame loss in total).
2. **Startup delay:** time interval between a peer decides to connect to a video session in mesh and begin its playback.
3. **End to end delay:** The time elapsed between the video frame originated from the source server and that video frame reaches at the client peer.
4. **Buffer map exchange period:** The time taken by peers to exchange their buffer maps to enable the other peers aware of the data availability in the neighborhood.

6. PERFORMANCE ANALYSIS

To measure and assess the impact of peer side buffer, buffer map exchange period over frame loss ratio, startup delay and end to end delay, the simulator was run under mesh based non cluster model by varying the number of peers with the observed parameters.

6.2 Network Performance of Non Cluster Model

Figure 2 depicts that varying the time period of buffer at peer side decrease the frame loss ratio and consequently yields better quality of received video to viewers. However, with growth in number of peers, the frame loss tends to increase.

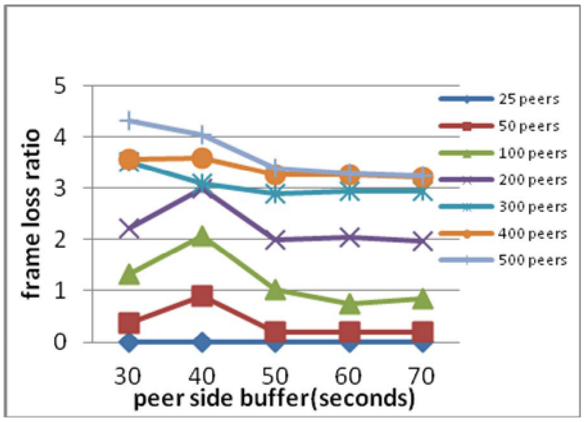

Figure 2: Impact of peer side buffer over frame loss (Non Cluster)

Figure 3 depicts that startup delay is constant irrespective of the increase of buffer time thus performance remains unchanged. However it is also observed that with increase in number of peers (25 to 200) initially startup delay increase but when network further increases (peers 300-500) it decrease.

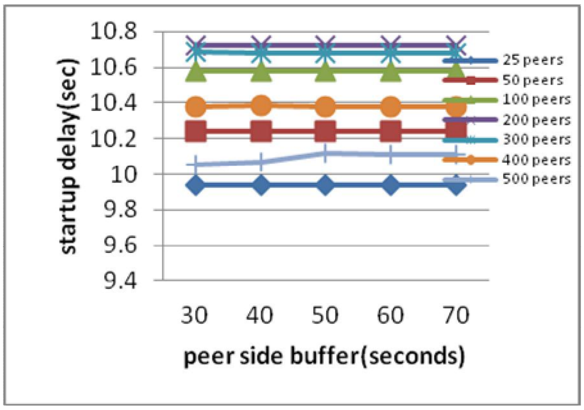

Figure 3: Impact of peer side buffer over startup delay

Figure 4 depicts that there is negligible impact on end-to-end delay by increasing the peer side buffer however due to increase in number of peers the end-to-end delay increase for this model.

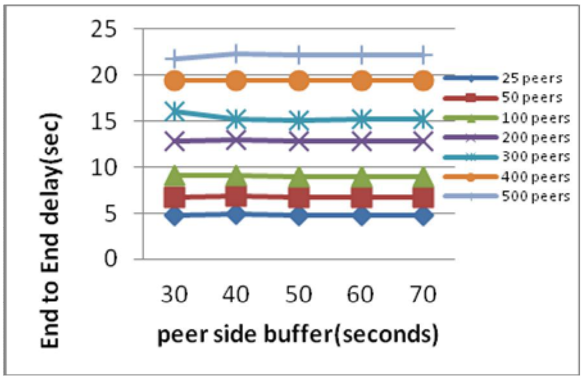

Figure 4: Impact of peer side buffer over end to end delay

Figure 5 shows that shorter period of buffer map exchange results in lower frame loss however increased buffer map exchange period tends to enhance frame losses.

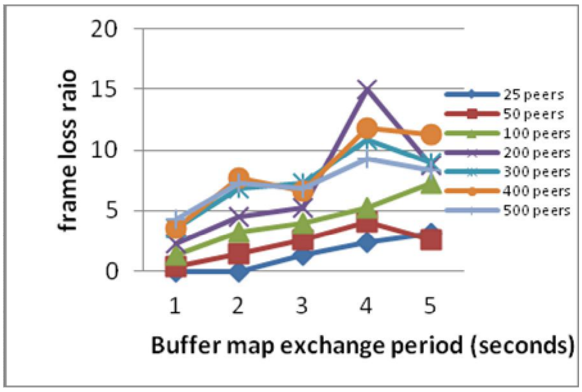

Figure 5: Impact of buffer map exchange period over frame loss (Non Cluster)

Graph at figure 6 shows that similar startup delay is maintained though network size changes. However startup delays exhibit rising tendency with increase in buffer map exchange period.

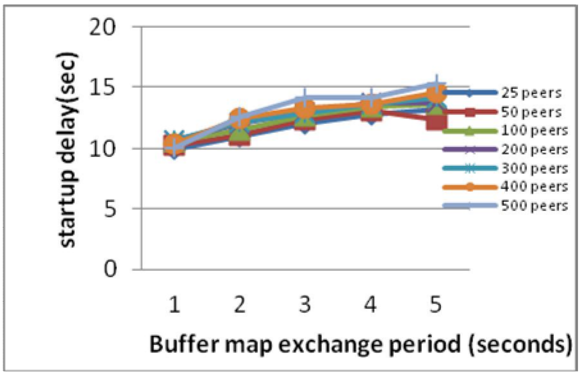

Figure 6: Impact of buffer map exchange over startup delay (Non Cluster)

Graph at figure 7 illustrate that end-to-end delay can be minimized with lesser period of buffer map exchange for better delivery of video. Further end-to-end delay appeared increase with growth in network size.

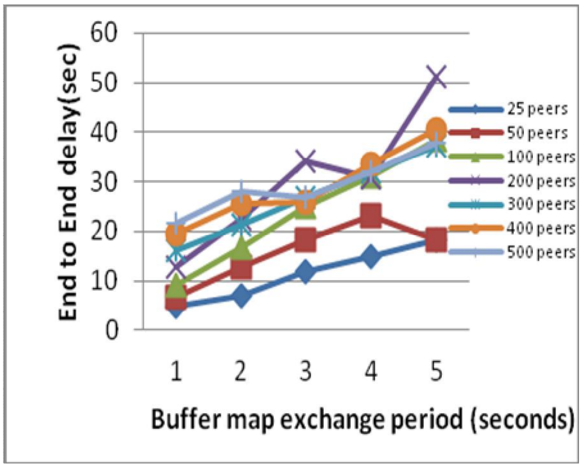

Figure 7: Impact of buffer map exchange over end to end delay (Non Cluster)

6.3 Network Performance of Cluster Model

To investigate the impact of peer side buffer, buffer map exchange period over frame loss ratio, startup delay and end to end delay under mesh based cluster model the simulator was performed by varying cluster sizes (fixed 300 numbers of peers) with the observed parameters.

Figure 8 depicts that increasing the time period of buffer results in decrease of the frame loss ratio. It is also seen that increasing number of clusters does not significantly degrade frame loss ratio.

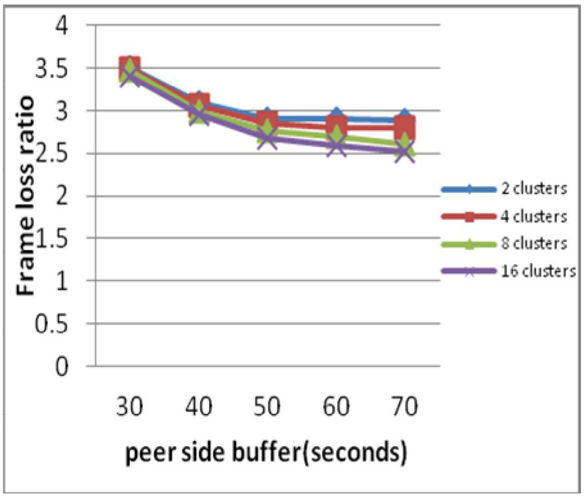

Figure 8: Impact of peer side buffer over frame loss(Cluster Model)

Figure 9 depicts that startup delay tends to decrease with increased cluster sizes. However startup delay is reduces for increase in size of cluster.

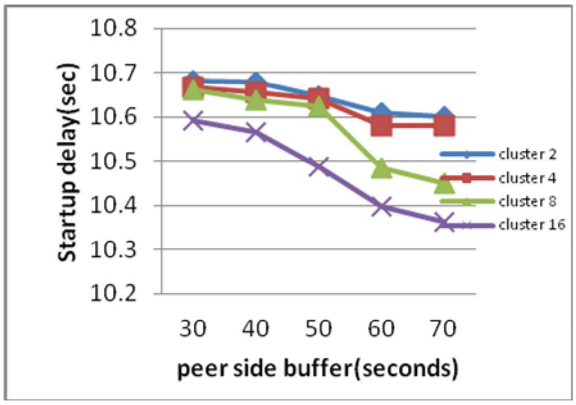

Figure 9: Impact of peer side buffer over startup delay (Cluster Model)

Figure 10 that there is reduction in end to end delay by increasing the peer side buffer which is more prominent initially and goes flat later. However it is nearly same for changes in cluster size from 2 to 16.

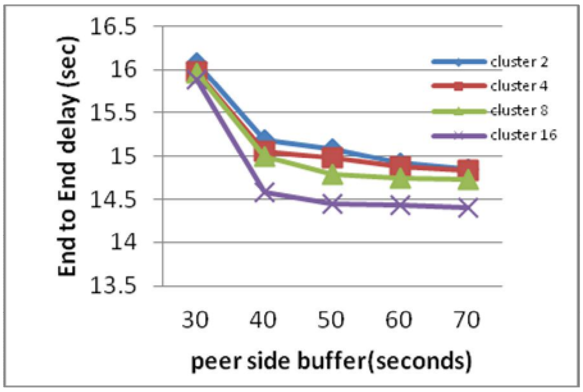

Figure 10: Impact of peer side buffer over end to end delay (Cluster Model)

Figure 11 that lower is the period of buffer map exchange better is performance on frame loss. However frame loss is also marginally lower only for largest (16) cluster size, but almost similar for other lower cluster sizes.

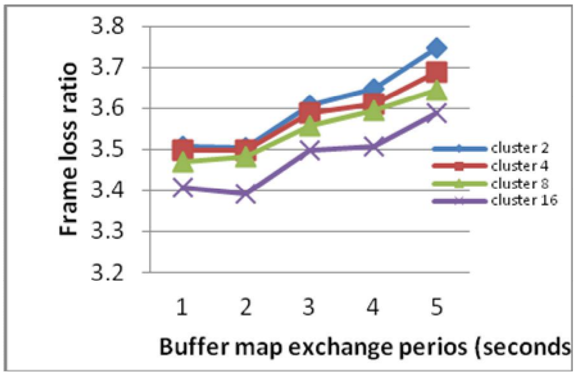

Figure 11: Impact of buffer map exchange period over frame loss (Cluster Model)

Graph at figure 12 that startup delay is maintained similar even with increase of cluster size and it tends to increase with increased buffer map exchange period.

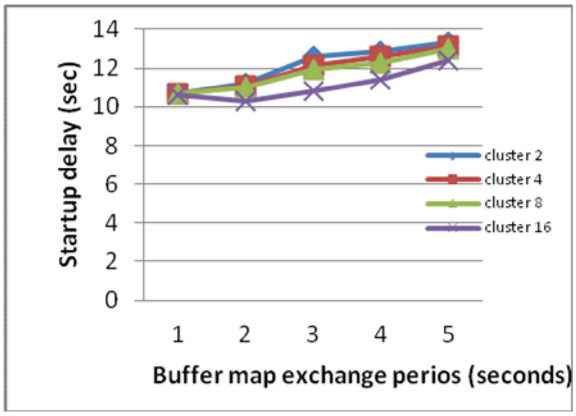

Figure 12: Impact of buffer map exchange period over startup delay (Cluster Model)

Graph at figure 13 that end-to-end delay can be minimized with lower period of buffer map exchange as well as increase in size of cluster for better delivery of video.

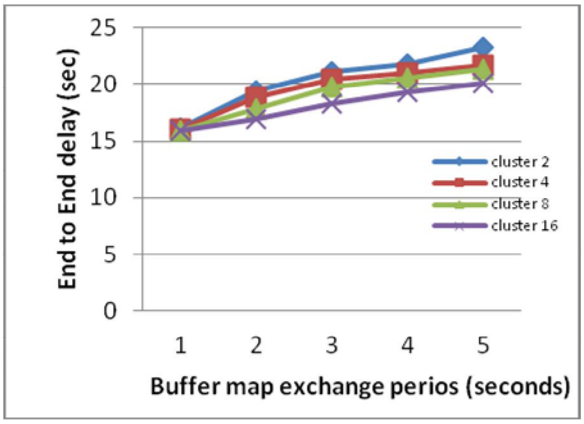

Figure 13: Impact of buffer map exchange period over end to end delay (Cluster Model)

7. CONCLUSION

This paper proposed a novel technique of clustering of peers in mesh based live video streaming for P2P VOD system. On the basis simulations under test setup following are extracted-

i) that buffer size at peer side improves significantly in both model but can be better managed due to creation of clusters in proposed cluster model. The impact of buffer in cases of clusters outperforms to reduce frame losses which cause less distortion in video quality.

ii)proposed cluster model outperform the non-cluster that the startup delay is minimized due to consideration of time bound limits of release buffer and playback buffers which bounds the startup delay almost constant in proposed model.

iii) that proposed cluster model minimizes delay due to formation of clusters which yields improvement in overall delay and consequently can provide prompt service to users increasing performance of VOD system (QoS becomes better).

iv) that both models perform better when the buffer map change period are shorter (1 second) but creates additional traffic by which frame loss, end to end delay increase. However to some extent it gets reduced in cluster model due to clustering of peers i.e. division of work.

v) It is also concluded that in non clustered model only one tracker manages all the activities performed by peers, where as clustered model is able to structure a hierarchy of trackers added, by which results improved a lot in management of peer churns.

vi) Further it is also inferred that the proposed model is able to provide late viewers (who joins late the live video) the video from starting and missing parts (of video), implies that it suits both live as well as stored streaming from users point of view, therefore it is serves functionally hybrid and most useful.

REFERENCES

[1] Xinyan Zang, Jiangchuan Liu, BO Li and Tak-Shing Peter Yem "Cool Streaming/ DONet: A Data-driven Overlay Network for Peer-to-Peer Live media Streaming", IEEE 2005.
[2] Shahzad Ali, Anket Mathur and Hui Zhang, "Measurement of Commercial Peer-to-Peer Live Video Streaming" 2006.
[3] Xiaojun Hei, Chao Liang, Jian Liang,Yong Liu and Keith W. Ross, “A Measurement Study of a Large-Scale P2P IPTV System”, IEEE Transactions on Multimedia, Vol. 9, No. 8, Dec 2007.
[4] Nazanin Magharei, Reza Rejaie and Yang Guo, “Mesh or Tree: A Comparative Study of Live P2P Streaming Approaches" Proc. of the IEEE International Conference on Computer Communications (InfoCom07), pp 1424-1432, 2007.
[5] Le Chang, “Video-on-Demand streaming on Peer-to-peer Networks”, Proc. International conference on Communications and Networking, IEEE 2007.
[6] A. Sentinelli, G. Marfia, M. Gerla, and L. Kleinrock, “Will IPTV ride the peer-to-peer stream?”, in Proc. IEEE Communications Magazine, vol. 45, no. 6, pp. 86-92, Jun. 2007.
[7] Y. Huang, T. Z. J. Fu, John C. S. Lui, and C. Huang, “Challenges, Design and Analysis of a Large-scale P2P-VoD System”, in Proc. ACM SIGCOMM’08 conference on Data communication, pp. 375-388, Aug. 17-22, 2008.
[8] C. WU and B. LI, “Exploring Large-Scale Peer-to-Peer Live Streaming Topologies”, in Proc. ACM Transaction on Multimedia Computing, Communications, and Applications, vol. 4, no. 3, pp. 19-23, pp. 1-25, Aug. 2008.

[9] H. Ketmaneechairat, P. Oothongsap, and A. Mingkhwan, "Smart Buffer Management for Different Start Video Broadcasting", in Proc. of ACM ICIS'09, pp. 615-619, Nov. 24-26, 2009.
[10] Yu-Wei chen and Yu-Hao Huang, "An Interactive Streaming Service over Peer-to-Peer Networks", Proc. of International Conference on Software and Computer Applications, 2011.
[11] R Bhadada, N C Barwar & R Bhansali , "Review of the Challenges to Remove Jitter and Packet Loss During Continuous Playback of Streamed Video Data in Video On Demand (VOD) System Receivers,, in International Journal of Computer Applications(IJCA) Special Issue on Electronics, Information and Communication Engineering, pp3-7,Dec 2011.

AUTHORS

Dr. N.C. Barwar is currently working as an Associate Professor in Department of Computer Science & Engineering, M.B.M Engineering College, J.N.V. University, Jodhpur. He received B.E. from MANIT Bhopal, M.E. in Digital Communication and Ph.D. from M.B.M. Engineering College, J.N.V. University, Jodhpur. He has published more than 40 papers in national and international conferences and journals and having teaching experience of more than 20 years at UG and PG level. His field of interest is Computer Networks, Multimedia, VOD, and Information theory etc.

Dr. Rajesh Bhadada was born on 12th April, 1964 at Bhilwara in Rajasthan (India). He graduated with Honours in Electronics and Communication from M.B.M. Engineering College, University of Jodhpur, Jodhpur, Rajasthan, India in 1985, earned M.Tech. in Computer Science and Engineering from the Indian Institute of Technology, Bombay in 1990. He was awarded Ph.D. by J.N.V. University, Jodhpur in 2008 in the area of VOD systems. He is a Fellow of the IETE and life member of the Computer Society of India. Presently he is serving as Associate Professor at Electronics and Communication Engineering department of M.B.M. Engineering College since 1987. Prior to joining teaching, he served as R & D Engineer at Uptron India Ltd., Lucknow, U.P.. He was a recipient of National Talent award by Govt. Of India in 1980. He is supervising researches for Ph.D. degree at Computer Science and Engineering as well as Electronics and Communication Engineering disciplines at his present workplace. He has specialisation and experience of over 23 years in the fields of Electronics and Communication, Computer Science & Engg as well as into IT. His research and teaching interest includes Computer communications and networking, Microprocessors and parallel processing, Computer graphics, Databases, Software Engineering besides Multimedia and Video On Demand Systems.