Analysis and Implementation of Peer to Peer Live Video Streaming System

Lekshmy A Sasi[1], Minu Lalitha Madhavu[2]

Department of Computer Science & Engineering, Sree Buddha College of Engineering, Alappuzha, Kerala, India

Abstract: ***The ever growing demand for video content attracted the attention of research from traditional server-based schemes to peer-to-peer systems for video dissemination. Users connect to each other, establishing a p2p overlay network on top of the Internet and give-and-take the video segments amongst themselves. The performance of a p2p system is characterized by its ability to disseminate the video content to all peers without errors and with the minimum possible delay. This constitutes a challenge since peers dynamically join and leave the overlay and also contribute different amounts of resources to the system. By considering these facts we have published a paper on P2P live video streaming [1]. This paper mainly focuses on the implementation and result analysis of our already published work.***

Keywords: P2P network, live video streaming, incentive mechanism

1. Introduction

Video quality, continuity and synchronization with the main source are the significant elements needed in every P2P live video streaming system [2]. However the reality or main problems in these systems are:

- The user needs to wait for a couple of minutes to begin the video.
- Less quality video.
- Most of the time video suffers from discontinuities.
- Video is asynchronous with main source.

As far as a user is concerned these are the main issues related to watching a live video. But in the implementation side the main challenges and issues are [3]:

A. *Throughput:*
The term throughput refers to data streaming rate in the system. Video segments must be streamed before its playback deadline and it in turns define the efficiency of the system. As far as a user is concerned it is the download rate. So a download rate greater than a playback rate is enough for fluently playing a video without any pauses or breaks.

B. *Transmission delay:*
This is the difference in playback points of different peers when they are watching the same live video. Videos in every peer must be synchronized with the live video in actual source.

C. *Startup delay:*
This refers to the time difference when user tunes to a particular video and the time when video is actually visible.

D. *Bandwidth utilization:*
No matter what scheduling algorithms are used, upload bandwidth or download bandwidth of the whole system is not infinite. So the maximum streaming rate of a system cannot exceed the available bandwidth.

E. *Inter ISP Traffic:*
Since most of the P2P streaming systems are not ISP friendly, their massive user base and huge traffic volume presents big challenge on ISP’s network capacities. When a peer in the overlay networks acquires some video segments from the video source ,it randomly connect to multiple peers, which can be either local or remote ,based on the content availability and also exchange data between different networks. So uncontrolled P2P video segment exchanges increases the inter ISP traffic.

F. *Free riders:*
Peers that do not offer their upload capacity are called free riders, because they try to acquire the video stream for free. The upload capacity of free-riders thus remains unused. This consecutively affects the overall performance of P2P network [4], [5].

The main purpose of our work is to provide the user a quality video within time and without any pauses or skips. The objectives are:

- Improve the system throughput
- Minimize transmission and startup delay
- Effectively use the available bandwidth by balancing the work load of data dissemination among peers.
- Minimize the inter ISP traffic
- Avoid Free-riders.

2. System Implementation

The different modules of this proposed system are:

- Overlay Construction and Maintenance
- Neighbor List Creation
- Segment Selection
- Peer Selection
- Incentive Mechanism

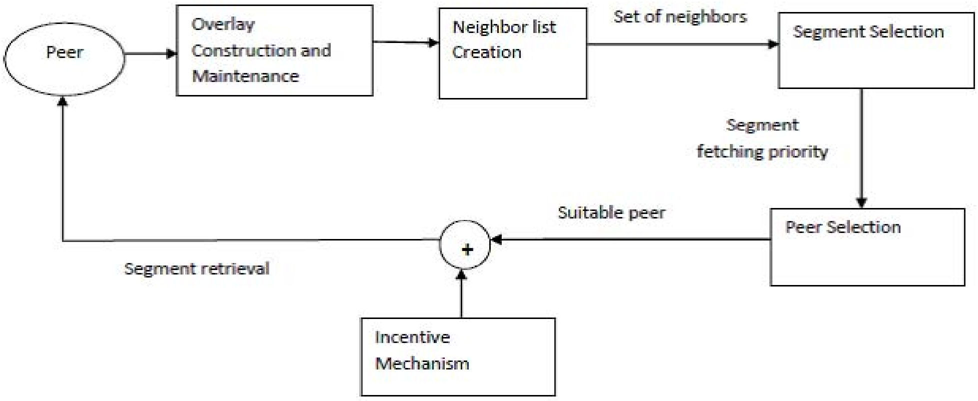


Figure 1: Architecture of P2P live video streaming system

Since we are mainly focusing on the implementation of these modules only such details are explained in this work and the detailed theoretical explanation of these modules are included in our previous work [1].

P2P live video streaming systems is implemented as two applications. An admin application considered as the tracker that keeps record of the registered peers and a client application which is installed in peers that are dynamically joining and leaving the network.

A. Admin

The main concept used in the implementation of admin side is the socket concept. A database named ‘*neighbourlist*’ is created to keep track of the active peers. The admin module mainly consists of five forms.

1) *Menu form*

This form provides links to all the other forms such as *sharefiles, peer creation, clientcheck* and *frmserversettings.*

2) *Frmserversettings form*

In the load event of this form hostname and IP address of the admin will be automatically loaded. For that purpose two built-in functions named *Dns.GetHostName()* and *Dns.GetHostAddress()* are used. Besides we can assign or modify the bandwidth of the admin. All these details will be written or stored to a text file named ‘*admin.txt*’ with the help of *StreamWriter()* function.

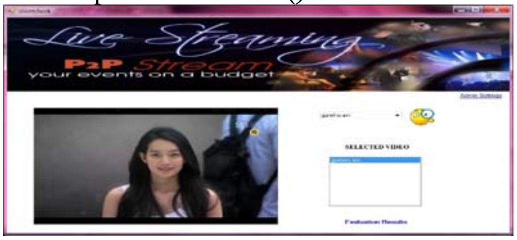


Fig 2. Clientcheck form

3) *Peercreation form*

In this system we give admin the authority to register a new peer with a username and password .We assume that client know this user name and password. ‘*Peercreation*’ form is designed for registering a new peer with a username, password and updating the fields of already registered peers. During the load event of this form IP addresses of all already registered peers will be loaded in the list box named ’*MEMBERS*’. If we want to modify any of information of a particular IP address, then click it and its corresponding details such as user name, user ID etc… will be loaded in the corresponding fields, so we can easily modify it. In the case of a new peer, the admin has to allocate a new password and user Id for the peer and all details will be stored in a text file named ‘*Peer.txt*’.

4) *Sharefiles form*

This form is for sharing the live video. The live video can be uploaded by the admin and a unique id will be automatically allocated for it. All the uploaded video will be loaded to a folder name ‘*fileshared*’ in the admin system and its details will be written in to a file named ‘*filelist.txt*’.

5) *Clientcheck form*

This is the most significant form of the admin module because most of the stuff needed to handle all peers in overlay network is coded in this form. The ’*clientcheck*’ form is shown in Fig 2. During the load event of this form, names of all videos that have been shared by the server will be loaded in a combo box and server can select a video from this list. As a result the video name will be shown in the list box. Video can be played in the admin form by clicking the video name in list box. Moreover the video segmentation is done in this form. With the help of FFmpeg library the video is divided in to segments of 10 seconds duration. A table named ‘*list*’ is used to keep track of the active peers. Focusing to the implementation side, mainly six sockets are used in this form.

1. Socket to receive messages such as login, logout from the clients.
2. Socket to receive file list request message. That means, when a client newly joins the overlay network it request name of the currently playing video from the server .For that purpose the server will be continuously listening for file list request message in this socket.
3. Socket to receive locality and ISP details of the joining peers.
4. Socket to listen the first segment request. I.e. when a peer joins the overlay network the first currently playing segment will be provided by the server. So this socket will be continuously listening for the first segment request from peers.

5. Socket to receive segment request from clients having only single neighbor but it is not having required segment.
6. A socket to listen the data, needed for evaluation of the whole system, from peers.

Connected to first three sockets a timer named "*conntimer*" is used. During every tick event of this timer, it checks whether any of these sockets received any data. If so, it checks the code word of the received message. Code word is appended in all the messages which are sent from client in order to differentiate among messages such as login, logout, file list request and video selected.

If the code word appended in the message received from client is "*logout*", the server will delete all the details of that particular peer from the ’list ’table. So that it won’t be allocated as neighbor to other peers.

If code word is "*login*" the server will extract the username and password associated with this code word. A text file named *’peer.txt’* contains the username and password of the already registered peers. So this text file will be searched to find the match for the received username and password .If a match is found a flag will be set, otherwise a message ‘*login failed*’ will be sent to the peer.

If the flag value is set,

- Server will assign a color to the peer using five color theorem [6].
- Update longitude, latitude and ISP of this peer in the locality.txt file.
- Message ‘*activated*’ along with the peer color will be sent to the peer.

When a peer select a live video to play, code word ’*selected*’ will be sent to the server. Now the server will extract the ISP, longitude ,latitude of this peer from the *’locality.txt’* file and select a list of peers that are currently watching that video using the neighbor selection method. This neighbor list is sent to the requested peer.

B. *Peer*

The different forms in peer side are

1) Login form

This form is used for the login of the peer system. The IP address of the peer system will be automatically loaded and the user has to enter user id, password, ISP and location information. ISP can be obtained either automatically or selecting from combo box. For entering the location information ‘*Google Map*’ is used and user can select location from this map. This user id and password will be sent to the server for verification. If it is a correct one then ‘*activated*’ message will be sent from the admin. This function is coded with the help of socket concept. When the Client program execute, it will establish a connection to the Server program and send request to the Server, at the same time it also receive the response from Server.

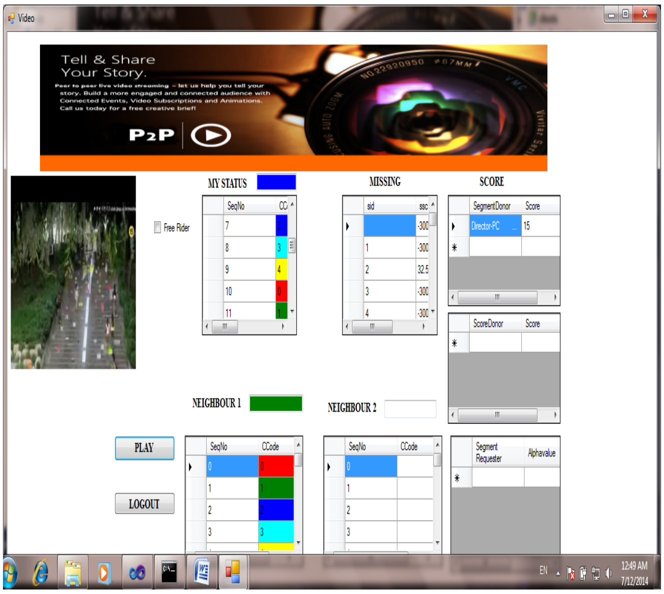
2) Search form

During the load event of this form names of currently playing live videos will be shown in the list view. Also user id along with its color will be displayed. The peer can select the live video from the list view. Another important feature implemented in this form is a timer which collects the status of the neighbors.

3) Video form

The live video will be played in this form. The design of form is shown in Fig. 3.Seven grid views are used. First grid view named *’MY STATUS’* will show the segment id and color of those segments that have been received by the peer. This color is calculated by taking mod 5 of segment id. The other two grid views named *’NEIGHBOUR 1’* and *’NEIGHBOUR 2’* shows the segment id and color of those segments that are present in the neighbors of this particular peer. The information about neighbor’s segment is obtained through window maps which will be exchanged periodically among neighbors. The grid view named *’MISSING SEGMENTS’* shows the id of those segments that are missing from this peer and its corresponding SSC value. The next three grid views are used to implement the incentive mechanism. First grid view for this purpose contains two fields. One field named *’SegmentDonor’* indicate the names of the peers that have given segments to this peer and other field named ’score’ indicates the score given to those peers for providing with segments. The second grid view also contains two fields, one field named *’ScoreDonor’* shows names of those peers that have given scores to this peer for providing them with segments and other field indicates the score provided by them. The third grid view indicates the names of those peers requesting segments from this peer, and their corresponding value. To simulate the effect of free rider a checkbox named *’free rider’* is also designed in this form. So when this check Box is ticked peer will act like a free rider and do not provide any segments to other peers. As a result of this score of this peer will get reduced. There are mainly 5 sockets used in this form.

1. Socket for sending window map to the neighbors.
2. Socket for receiving window map from other peers.
3. Socket to request segments from other peers.
4. Socket to receive segments from other peers.
5. Socket to receive segment request from other peers.


Video
Tell & Share
Your Story.
Peer to peer live video streaming – let us help you tell your story. Build a more engaged and connected audience with Connected Events, Video Subscriptions and Animations. Call us today for a free creative brief!
P2P
MY STATUS
MISSING
SCORE
NEIGHBOUR 1
NEIGHBOUR 2
PLAY
LOGOUT
EN
12:49 AM
7/12/2014

□ Free Rider

SeqNo	CC
7	2
8	3
9	4
10	0
11	1

sid	ssc
	-300
1	-300
2	32.5
3	-300
4	-300

SegmentDonor	Score
Director-PC ...	15
*	

ScoreDonor	Score
*	

SeqNo	CCode
0	0
1	1
2	2
3	3

SeqNo	CCode
0	
1	
2	
3	

Segment Requester	Alphavalue
*	

Figure 3: Video form

3. Results and Discussion

From the grid views named *'MY STATUS'*, *'NEIGHBOUR 1'*, *'NEIGHBOUR 2'* of video form it is clear that the video segment streaming rate is very high and it is greater than the playback deadline. A graph drawn with number of segments as 'Y' axis and interval as 'X' axis is shown in Fig.4. To draw this, MS chart control in Visual studio 2010 is used. In the *'video form'* of peer application a timer named *'evaluation timer'* is placed and it periodically sends information about the availability of segments to the admin application. Admin application stores this information in a table named *'evaluation'* and with this graph is produced. From graph it is clear that when number of neighbors is 0, the total segments with in a particular interval are low. When a new neighbor comes, the total segments in a particular interval are increased. Moreover the segment distribution rate is great. This is one of the aspects that contribute to the fluent playing of video in peer, which in turn means the system achieved high throughput.

The next concern in the problem description was the transmission delay. Each video segment is of 10 seconds duration. When a peer selects video, the first currently playing segment is sent from the server and the maximum delay that can occur (peer side) in playing is 9 s. i.e., The server may be playing the 10th second of the currently playing segment, but the client has to start playing from the 1st s of that video segment. Other segments can be fetched very fast from neighboring peers because of the neighbor selection policy and peer selection policy. As a result transmission delay is minimized.

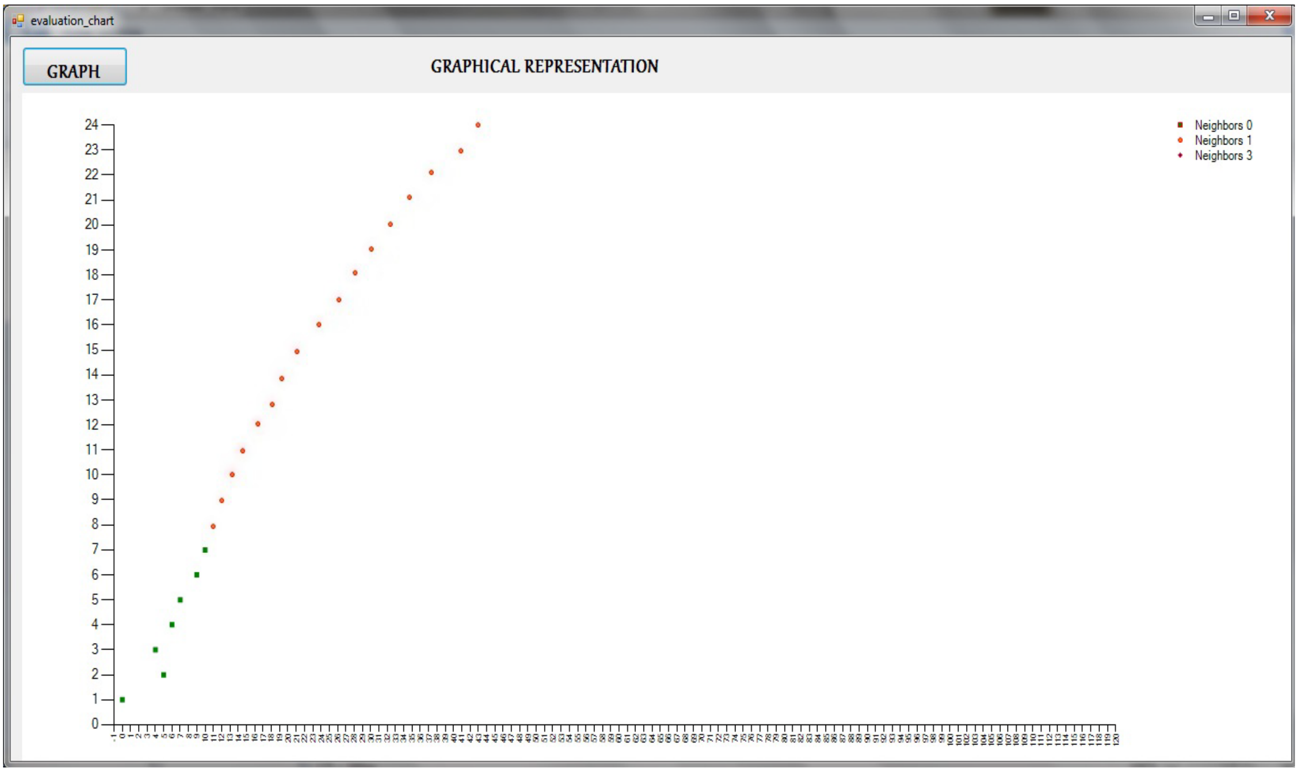


Figure 4: Evaluation graph

From the grid view for the incentive mechanism it is clear that the score of the free riders get reduced and this in turn reduces their value. Thus every peer in this network is compelled to upload the segments and this improves the efficiency and performance of the system.

For segment selection, by considering the factors such as urgency, rarity and freshness the segment distribution becomes fast and this minimizes the startup delay. Besides it avoid the pauses and skips in video. With the help of peer selection module the work load of segment distribution can be evenly distributed among the peers and this causes the effective use of available bandwidth.

4. Conclusion and Future Work

In this work we have studied the different problems in designing a high performance P2P live video streaming system and proposed a neighbor selection policy and an enhanced peer selection policy based on ISP and locality information which reduces the inter ISP traffic and further improves the system throughput by effective utilization of available bandwidth. Additionally this method reduces the startup delay. Moreover we proposed an incentive mechanism which evades the free-riders and this in turn coerce all peers to contribute segments .As a result of this segment distribution becomes faster and throughput improves. All these modules have been implemented successfully and live video is played in peer systems without any pauses or disruptions. As a future work we would like to consider more factors for segment selection and also like to implement this in more number of peers.

REFERENCES

[1] Lekshmy A Sasi, Minu Lalitha Madhavu, “Peer to Peer Live Video Streaming”,International Conference on Emerging Trends in Engineering and Technology, March- April 2014.

[2] Lekshmy A Sasi, Minu Lalitha Madhavu,”A Survey on Peer to Peer Video Streaming Systems”, vol. 3. International Journal of Research in Computer Engineering and Electronics ,July 2014,in press.

[3] M. Castro, P. Druschel, A. M. Kermarrec, A. nandi, A. rowstron, and Singh, “A Splitstream: Highbandwidth Multicast in Cooperative Environments,” In Proceedings of ACM SOSP, 2003.

[4] C. Dana, D. Li, D. Harrison, and C. Bass, “Bittorrent assisted streaming system for video-on-demand,” in International workshop on multimedia signal processing (MMsP), 2005.

[5] X. Zhang, J. Liu, B. Li, and Y. P. Yum, “CoolStreaming/DONet: A Datadriven Overlay Network forPeer-to-Peer Live Media Streaming,” in Proc. IEEE INFOCOM, 2005, pp. 2102-2111, 2005.

[6] X. Tu, H. Jin, J. Cao, S. Guo, L. Zheng, and Z. Lv, “An EfficientData Scheduling Scheme for p2p Storage Constrained IPTV System,” IEEE Transactions on Systems, Man and Cybernetics:Systems,vol.43,no.2, 2013.