

A Social-Network Based Multichannel Online Video Streaming Mechanism over Peer-To-Peer Networks

Sangeeta B¹, Deepa K.S²M. Tech Student, Department of Computer Science, AMC Engineering College, Kalkere, Bengaluru, India¹Assistant Professor, Department of Computer Science, AMC Engineering College, Kalkere, Bengaluru, India²

ABSTRACT: In today's distributed (P2P) live streaming frameworks, if nodes need to share recordings between one associate to another companion they shape a shared overlay. In existing live streaming applications, there are several channels that are telecast to a great many clients in the meantime. This expanded number of channels accelerates clients' wish of observing increasingly channels regular and few channels in the meantime. In any case, to watch two channels all the while a node needs to stay in two diverse channel overlays this will bring about an overwhelming burden on concentrated server. (1) To diminish the overwhelming burden on the unified server, (2) To lessen the channel exchanging postponement and (3) to permit nodes to watch two channels in the meantime in a same window we propose a Social-system Aided proficient live spilling framework (SAVE). To accomplish these points SAVE presents two primary plans they are channel grouping plan and friendlist plan. To develop the channel bunch for channels SAVE considers the intrigued channels of every nodes and their regular associations between those channels and the single channels are gradually assembled into channel groups. Also, to develop the friendlist SAVE requests that nodes enter their intrigued channels and watching time of those channels physically. SAVE additionally proposes the channel closeness-based piece pushing technique and limit based chunk supplier determination methodology to expand the framework execution. Trial results demonstrate that SAVE performs superior to anything different frameworks.

KEYWORDS: Peer-to-peer (P2P) live streaming, P2P networks, social networks.

I. INTRODUCTION

To watch channels or to stream online recordings there are numerous current live spilling applications like PPLive and UUSEE. Since they utilize decentralized nature of distributed systems (i.e. nodes are not relying upon the unified server), a large number of clients are intrigued to utilize those applications. Since they have hundreds to a huge number of directs in their present application, they are telecasting those stations to a huge number of clients in the meantime. On the off chance that nodes need to share recordings between one associate to another associate they frame a distributed overlay for the specific channel.



Fig.1. Multichannel interface.

In a multichannel watching mode, clients can watch two directs in a same window. What's more, in a window one primary perspective will be there and one auxiliary perspective will be there, this is called as PIP (i.e. Picture in

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 10, May 2016

Picture). Utilizing this PIP client can switch both the channels between the principle view and the optional view rapidly and uninhibitedly.

Be that as it may, the majority of the current live spilling applications bolster single channel review i.e. they will permit clients to watch one channel at once. Just few live spilling applications support multichannel seeing i.e. they will permit clients to watch two directs progressively in one window. Be that as it may, to watch two channels they require two channel overlays. For instance, if a client is watching two channels he needs to keep up two channel overlays to switch between each other.

In spite of the fact that there is a current application called PPStream which bolsters picture in picture furthermore utilizes this technique. In PPStream if a hub is watching two channels then he needs to keep up two channel overlays. Due to this maintenance cost increases.

Finally SAVE aims at the following

- Low overhead
- Quick reaction
- Light server load

The remainder of the paper is organized as follows. Section II describes the related work. Section III describes the design of SAVE. Section IV shows the experimental results. Section V concludes the paper with remarks on future work.

II. RELATED WORK

P2P Live Streaming Channel Overlays: P2P live spilling conventions fall into four classes: tree-based, meshbased [3], half and half structure joining both work and tree structures and disseminated hash table (DHT) - based structure [2].

Multichannel P2P Live Streaming Techniques:

Chen et al. [5] proposed OAZE, where every companion keeps up associations with other physically close companions in a specific number of channels, which its related client is prone to watch. At the point when a hub needs to switch to a channel, it tries to discover neighbors interfacing with the objective channel for the switch. Be that as it may, a hub's neighbors may not interface with the objective channel, so OAZE can't give a high likelihood for a hub to discover a neighbor to change to its objective channel.

Interpersonal organization Aided Video-on-Demand Systems: Several interpersonal organization construct video-in light of interest frameworks have been proposed. They advance proposed NetTube [1] that uses the connections between related recordings to create an informal community of hubs, and afterward utilizes an interpersonal organization helped prefetching methodology to accomplish smooth move between video playbacks.

III. DESIGN

A. Overview of SAVE

Fig. 2 demonstrates an abnormal state perspective of the structure of SAVE. The server node is the focal point of the whole system. At first, all nodes in every channel shape an overlay. In SAVE, every channel overlay has a channel head indicated by, which is a steady node with the most noteworthy limit and longest lifetime staying in the channel. SAVE has two principle plans: channel grouping and friendlist.

B. Channel grouping

We utilize channel closeness of two channels to mirror the recurrence of cooperations between these channels, i.e., the late propensity of nodes to switch between or watch both channels. Such an inclination can be assessed by three elements: 1) the age (i.e., freshness) of the node's exchanging or multichannel watching movement on both channels; 2) the time period that the node stays in both channels; and 3) if both the directs are in the node's intrigued channel list. We acquaint how with consider these components to ascertain the channel closeness.

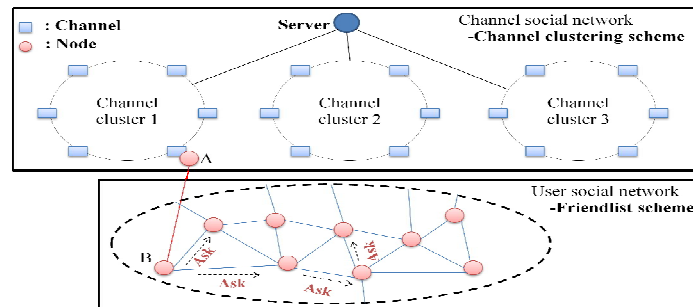


Fig.2. High level view of the SAVE structure

As appeared in Fig. 3(a), for a node's changing action from channel x to channel y , we characterize age as the time slipped by since the exchanging. We characterize $t(x)$ as the time interim that the node stays in channel x before exchanging. Also, as appeared in Fig. 3(b), for a node's multichannel watching movement on both channels, we characterize age as the time slipped by since the multichannel watching is begun, and characterize $t(x)$ and $t(y)$ as the time interim that the node stays in both channels; note $t(x)=t(y)$. Whenever $t(x)$ achieves a specific quality, we consider the node is really intrigued by watching channel x .

In like manner, we predefine an edge T_s , and characterize parameter $I(x) = 1$ when $t(x) \geq T_s$; generally $I(x) = 0.1$. In SAVE, every node has a profile that rundowns its intrigued channels indicated by the client. On the off chance that both channels x and y are in the node's intrigued channels, we consider the exchanging non coincidental, and set the estimation of parameter γ to 1. Else, we set $\gamma=0.1$ to minimize the impact of unplanned exchanging exercises. The scale for parameter I and γ is flexible.

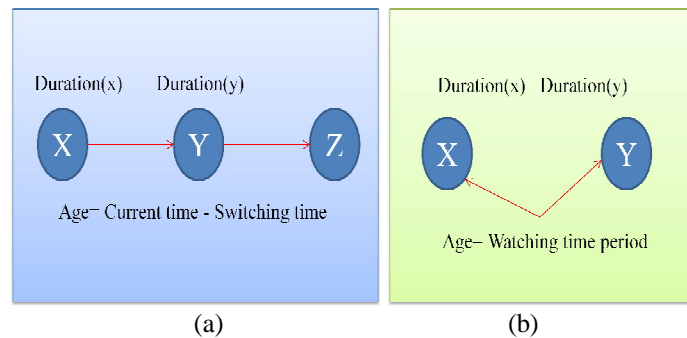


Fig.3. Illustration of the parameter definitions. (a) Channel switching. (b) Multichannel watching.

The channel head of channel y keeps a record of channel watching and exchanging exercises of nodes in channel y (denoted by Ω) and computes the channel closeness between channels x and y by

$$C(x, y) = \sum_{\Omega} \frac{I(x)I(y)}{\omega^{age}} \cdot \gamma \quad (1)$$

Where $\omega > 1$ is a scaling parameter, which exponentially decreases the freshness of exchanging and multichannel watching exercises. In this manner, the estimation of $C(x, y)$ is in the scope of $[0, 1]$. The closeness of two channels can be viewed as the heaviness of a connection associating them in the informal organization diagram. The channel bunching is the procedure of collection channels with high-weight joins.

$$v(cr_i, cr_j) = \left\{ \sum_{x,y \in \Omega} c(x, y) - \sum_{x \in cr_i, y \in cr_j} c(x, y) \right\} / |cr_i|$$

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 10, May 2016

$$S_{cr_i} = P_i \cdot V_i^T$$

SAVE intends to produce groups so that the quantity of intracluster communications is boosted and the quantity of intercluster connections is minimized.

Algorithm 1: Centralized channel grouping algorithm

1. $G = V \cup s$;
2. Connect s to V to generate an expanded graph $G(V, E)$;
3. **for** all nodes $v \in V$ **do**
4. Connect v to s with an edge of weight w ;
5. Calculate the minimum cut tree T of G ;
6. Remove s from T ;
7. Divide G to clusters;
8. **Return** all connected sub graphs as the clusters of G ;

Algorithm 2: Decentralized channel grouping algorithm executed by cluster head

1. Calculate V_i and S_{cr_i} ;
2. for each interacted channel cluster $cr_k \in (\Theta - cr_i)$ do
3. $Cr_{(i,k)} = cr_i \cup cr_k$;
4. Ask for information from h_{cr_k} ;
5. for each channel cluster $cr_a \in (\Theta - cr_i - cr_k)$ do
6. Calculate $V(cr_{(i,k)}, cr_a)$ in $V_{(i,k)}$;
7. Calculate $P_{(i,k)}$;
8. Calculate $S_{cr_{(i,k)}} = P_{(i,k)} \cdot V_{(i,k)}^T$;
9. if $S_{cr_i} < S_{cr_{(i,k)}}$ then
10. // $cr_{(i,k)}$ is more stable than cr_i ;
11. Return cr_k ; //return the selected cr_k ;

C. Friendlist Construction

Today's live gushing applications normally rundown various interest labels for channels. Spare solicitations clients to fill their advantage labels physically when they at first join in the framework and to intermittently redesign their labels. Fig. 4 demonstrates a node's profile in light of its own direct watching exercises in SAVE.

D. Effective Multichannel Video Streaming

At the point when a node at first joins in SAVE, it solicitations to suggest nodes in its coveted channel. Node then joins in the channel overlay by interfacing with the prescribed nodes for piece sharing. In a group, channels could be converged into one overlay or spanned. Since the diverts in a blended overlay are near each other (i.e., nodes regularly lead progressive or multichannel watching exercises on these channels), when a node in a consolidated overlay needs to switch channel or watch multi channels, it has a high likelihood to locate its asked for lumps from its overlay utilizing the first piece seek calculation.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 10, May 2016

| IP Address | Similarity | Profile | Entry Creation Time | | | | |
|---------------|-------------------------|-------------|---------------------|---------------|---------------|---------------|---------------|
| ↓ | | | | | | | |
| Interest Tag | Channel | Watch time | Active Vector | | | | |
| News | BBC, NDTV | 12 hours | 00010010 | | | | |
| Sports | TEN Sports, Star sports | 33 hours | 00011010 | | | | |
| ... | ... | ... | ... | | | | |
| Active Vector | | | | | | | |
| [0:00-3:00] | [3:00-6:00] | [6:00-9:00] | [9:00-12:00] | [12:00-15:00] | [15:00-18:00] | [18:00-21:00] | [21:00-24:00] |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

Fig. 4: Record for a friend in the friendlist.

E. Limit Based Chunk Provider Selection

A node's ability speaks to the quantity of lump solicitations it can simultaneously serve. To evade the inertness for the accessible limit questioning, the nodes in every channel overlay can occasionally report their accessible abilities to their channel head, which encourage shows this data to all nodes in the channel overlay. Along these lines, a channel head or a companion can specifically prescribe the nodes with the most elevated accessible limit. By associating with high-limit nodes as piece suppliers, the lump requester can have a superior watching knowledge.

F. Structure Maintenance in Node Churn

Node stir is for the most part about the node joins and takeoffs from the framework, specifically the nodes are getting logged off or on the web. Spare needs to keep up its structure in node beat. To guarantee there is dependably a head node in every channel, before a channel head leaves, it chooses another head node and exchanges the greater part of its data to the new head.

G. Channel-Closeness-Based Chunk-Pushing

Instructions to completely use the constrained store of every node to diminish channel seeing startup postponement is a test. At the point when the reserve utilized for a channel achieves 660 kB, the store hit rate about achieves 100%.

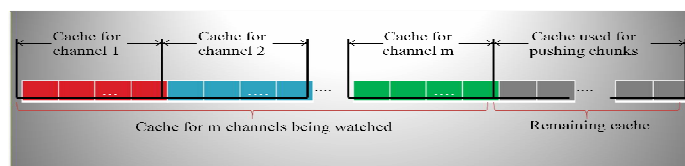


Fig.5. Cache allocation for different channels in a node.

IV. EXPERIMENTAL RESULTS

Figures show the various experimental results. Fig.6 shows the various graphs for (a) channel switching delay, (b) Server load, (c) Distribution of resolved switching requests, (d) Effectiveness of friendlist, (e) Clustering cost with respect OAZE on planet lab.

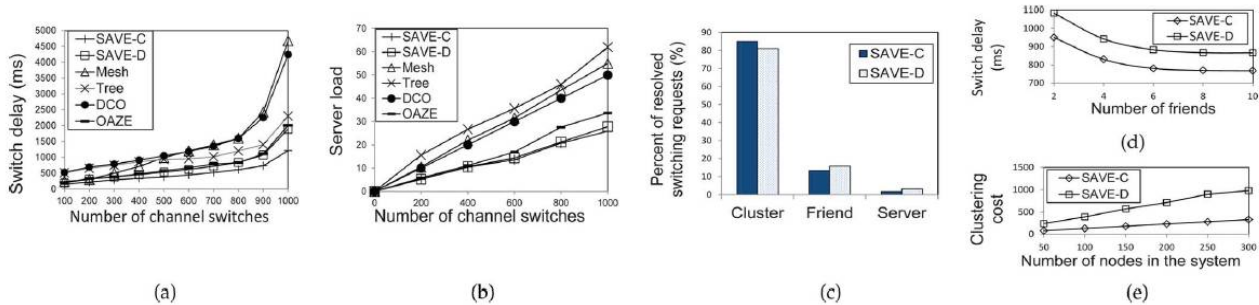


Fig.6. Experimental results from the PlanetLab testbed. (a) Channel switch delay. (b) Server load. (c) Distribution of resolved switching requests. (d) Effectiveness of friendlists. (e) Clustering cost.

Fig.7 shows the impact of friendlist on planet lab with respect to channel switching delay and server load.

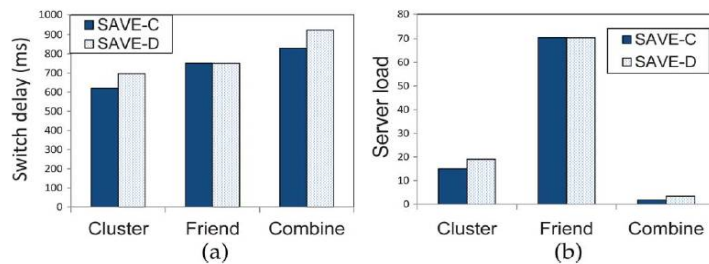


Fig.7. Impact of friendlist on PlanetLab. (a) Channel switch delay. (b) Server load.

V. CONCLUSION

In this paper, we propose SAVE, an interpersonal organization helped productive P2P live gushing framework. SAVE backings progressive and different channel seeing with low switch postpone and low server overhead by upgrading the operations of joining and exchanging channels. SAVE considers the verifiable channel exchanging exercises as the social connections among channels and groups the as often as possible communicated channels together by consolidating overlays or constructing spans between the overlays. Our future work lies in further decreasing the expense of SAVE in structure upkeep and node correspondence. Likewise, we will outline calculations for group division and decentralized bunch head race.

REFERENCES

- [1] X. Cheng and J. Liu, "NetTube: Exploring social networks for peer-to-peer short video sharing," in *Proc. IEEE INFOCOM*, 2009, pp. 1152–1160.
- [2] H. Shen, Z. Li, and J. Li, "A DHT-aided chunk-driven overlay for scalable and efficient peer-to-peer live streaming," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 11, pp. 2125–2137, Nov. 2012.
- [3] Y. Chu, S. Rao, and H. Zhang, "A case for end system multicast," in *Proc. ACM SIGMETRICS*, 2000, pp. 1–12.
- [4] J. Liang, B. Yu, Z. Yang, and K. Nahrstedt, "A framework for future internet-based TV broadcasting," in *Proc. IPTV*, 2006, pp. 1–6.
- [5] Y. Chen, E. Merrer, Z. Li, Y. Liu, and G. Simon, "OAZE: A networkfriendly distributed zapping system for peer-to-peer IPTV," *Comput. Netw.*, vol. 56, no. 1, pp. 365–377, 2012.
- [6] X. Liao, H. Jin, Y. Liu, L. M. Ni, and D. Deng, "AnySee: Peer-to-peer live streaming," in *Proc. IEEE INFOCOM*, 2006, pp. 1–10.
- [7] J. Liu, S. G. Rao, B. Li, and H. Zhang, "Opportunities and challenges of peer-to-peer internet video broadcast," *Proc. IEEE*, vol. 96, no. 1, pp. 11–24, Jan. 2008.
- [8] F. Dobrian, V. Sekar, I. Stoica, and H. Zhang, "Understanding the impact of video quality on user engagement," in *Proc. ACM SIGCOMM*, 2011, pp. 362–373.