An Efficient Online Video Streaming Mechanism Over Peer-to-Peer Network

Shreedevi Herur
M.Tech Scholar
Department of Computer Science & Engineering
Don Bosco Institute of Technology

Yashashwini B M
Assistant Professor
Department of Computer Science & Engineering
Don Bosco Institute of Technology

Abstract— **Video sharing has been an undeniably well known application in online informal communities. Be that as it may, it is not just excessive as far as server transfer speed and capacity additionally not versatile with the taking off measure of clients and video content. The associate helped Video-on-Demand (VoD) method, in which partaking peers help the server in conveying video content has been proposed as of late. Tragically, recordings must be spread through companions in OSNs. Along these lines, current VoD works that investigate grouping nodes with comparative premiums or close area for elite are problematic. In distributed (P2P) live gushing frameworks, which allude to content convey live and sharing video, henceforth every hub required to shape an overlay. A node triggers to watch another channel relies on brought together server. Live gushing applications make clients to watch different channel at the same time, which if broadly utilized, server postures substantial burden. To enhance proficient live spilling frameworks, we proposed SAVE. SAVE gathers data of nodes interests, channel's watching time and channel associations. It shapes an overlay between nodes with regular communication; distinguish nodes comparable intrigue and watching times. Furthermore fabricates spans between overlay of channel with less incessant connections. We have planned database structure for SAVE frameworks utilizing MYSQL and developed formats for client and administrator login, video transfers and made friendlist utilizing Net Beans IDE 7.**

Keywords— Video-on-Demand, P2P Networks, Online Social Networks, Live Streaming, SAVE.

I. INTRODUCTION

The most prevalent Online Social Networks (OSNs) like Facebook, Twitter, Google+, and so forth., are currently among the most mainstream locales on the Web. OSNs sets up social connections and these systems gives an intense method for sharing, sorting out, and discovering content. Worldwide there are 950 million Facebook clients are there and every day 500 million clients logon to Facebook and video transfers are 300 million every day. It has more than 1.32 billion month to month clients not at all like other video sharing frameworks [5] like BitTorrent and YouTube, which are composed around substance as opposed to clients. In OSNs clients build up the social associations with the companions in certifiable and overhaul their profiles and substance share their photographs, recordings, and notes to their own pages. Video sharing has turned into a most prominent among clients to impart their fascinating recordings to their companions in OSNs. Facebook is the second-biggest online video seeing stage as indicated by comScore Releases in August 2010. The aggregate time spent on video seeing on Facebook expanded 1,840% year-over-year, from 34.9 million minutes to 677.0 million minutes from October 2008 to October 2009. Video viewers have been likewise expanded by 548% amid the same timeframe and aggregate number of streams developed by 987%. OSNs have become a platform for catching up with friends, for personal expression and for sharing variety of content. Several approaches have been proposed for automation of shopping mall. However most of them focused on one aspect of the problem. In [1] **:** Peer-to-peer (P2P) content distribution is able to greatly re- duce dependence on infrastructure servers and scale up to the demand of the Internet video era. However, the rapid growth of P2P applications has also created immense burden on service providers by generating significant ISP-unfriendly traffic, such as cross-ISP and inter-POP traffic In [2], Video-on-demand in the Internet has become an immensely popular service in recent years. But due to its high bandwidth requirements and popularity, it is also a costly service to provide. We consider the design and potential benefits of peer-assisted video-on-demand, in which participating peers assist the server in delivering VoD content.. In [3], Video-on-Demand streaming on Peer-to-Peer(P2P) networks has been an emerging technique in recent years.

In late research, P2p Vod administration has progressively gotten consideration. A few on interest P2p feature organizations have been proposed through broad dissemination and functionalities of the Vod administrations examination [2]-[4]. In addition, there is likewise a large number of notices installment models (i.e., cost-for every activity, cost-for every click, cost-for every impression, cost-for every download, and expense for every guest) in feature on-interest or promoting underpinned Vod channels that are, no doubt conveyed [4].

Most of P2P live streaming applications such as P2PIPTV & PPstream are centralised server. The success of these applications made the need of decentralized server, which posses to reduce load capacity on centralised server. Each node contact the centralized server for every new channel wants to watch. The support of successive and simultaneously watching of multiple channels at a time is difficult in current P2P live which allow users to share stream in one channel. A node watching multiple channels which required forming multiple P2P overlay and it maintains cost is increased. As a node opens more channels, it leads to heavy burden on centralized server and delay response makes inefficiency in P2P live streaming systems. SAVE proposed to improve the efficiency of live streaming systems. The design of this project is based on utilization of social network. The two main schemes of SAVE system are: channel clustering and friendlist.

Channel clustering scheme: The vast majority of nodes watch comparable interest channel at same time and node watching restricted to little number of channels. Henceforth SAVE performs channel grouping with incessant connections of channels. It shapes overlay between channels with continuous cooperations and develops spans between overlay of channel with less incessant communication. Along these lines, the progressive or multichannel viewing of clients in its present overlay or extensions shape new overlay with meddling server.

Freindlist scheme: A node reaches its destination node with few numbers of steps which implies that a node in channel can knows another channel in few steps through friend connections.

Thus every node in SAVE maintains friendlist to save the details of nodes sharing common channels interest and watching time period. If a node wants to watch a channel that is not in present overlay, it refers to friendlist to switch nodes in desired overlay of channel.

II. RELATED STUDY

In Online Social Networks video seeking usefulness is not gave not at all like other video sharing frameworks (e.g. YouTube) and recordings must be shared by companions and companions of companions. So we take into contemplations like social separation, physical separation as the vast majority of the recordings watched by dear companions and have a tendency to be situated in same area, and video length. As the vast majority of the recordings saw are short length recordings. Normal time spent on Facebook per client is 20 minutes. Cheng et al., [2] proposed a model called NetTube a novel to show distributed helped conveying system which examines the grouping in interpersonal organizations for short length video sharing. Their work concentrates on arrangement of key outline issues to understand the system, a bi-layer overlay, a productive indexing plan and a pre-bringing technique utilizing interpersonal organizations. internet Video-on-Demand be productive? Inside and out. Video-on-Demand applications have turning into an undeniably well known an in web in the course of the most recent couple of years. What's more, it has turned out to be exorbitant because of its high data transmission prerequisites and notoriety. Their work concentrates on configuration and advantages of associate helped Video-on-Demand, where companions can help the server in conveying VoD content. The help can be given in such a way, to the point that it gives the same client quality and experience as that of customer server dissemination components. Additionally they concentrate on the single-video methodology, whereby an associate can just redistributes a video that has been right now viewing.

P2P Live Streaming Channel Overlays: P2P live gushing conventions fall into four classes: tree-based [21]–[24], meshbased [25]–[32], cross breed structure joining both work and tree structures [15], [33]–[37], and conveyed hash table (DHT)- based structure [20]. Tree-based techniques convey video content by means of push system, in which parent nodes forward got pieces to their kids. Early proposed tree-based methods such as Narada [23] and multicast [22] rely on a single tree structure, which is vulnerable to churn. Recent works [24], [34] build multiple trees, in which leaf nodes join in several trees to improve the system resilience to churn. Mesh-based methods [26]–[28] connect nodes in a random manner to form a mesh structure. Each node usually serves a number of nodes while also receiving chunks from other nodes. eQuus [29] further facilitates clustering and locality-aware mechanisms, while other works [30]–[32] introduce improved packet scheduling protocols. Mesh-based methods are resilient to churn, but generate high overhead by frequent content publishing.

Shenet al., [4] given a mechanism called DHT-aided chunk-driven overlay for scalable and efficient P2P live streaming. Due to internet-based video streaming applications are gaining more popularity, attracting millions of users every day. This growth of users poses problems like high video Quality of Service (QoS), availability, scalability and low-latency challenges to peer-to-peer assisted live video streaming systems. Tree-based systems with low-delay are vulnerable to churn, while mesh-based systems are churn-resilient and have a high delay and overhead. They address these problems by introducing a DHT aided chunk driven overlay.

DHT aided Chunk Driven Overlay selects stable nodes for efficient chunk sharing we call these nodes coordinators. Chunks are transmitted in top-down manner with decreasing bandwidth. **Node Join:** Server keeps track of number of nodes in the system and selects a node depending on its active life period makes it as coordinator (stable node) or joins it to the particular coordinator.

Node departure and failure: Node can either leave informing to the neighbor nodes or abruptly it can leave the system. If node failure happens in the system certain time out period will be noticed and coordinator will remove that node from DHT table if coordinator node failed than after communication failure with the coordinator server assigns new coordinator to the system.
In Figure 1, dark circled nodes are stable nodes(Coordinator).Each stable node has a ID and Name. Video chunks with ID 001 and name CNN0001 are in the first entry. Stable node N4 is the owner of ID 001, ID 002, and ID 004, it stores the video chunks with these IDs. (Indices of specific chunk of different nodes gather in same coordinator.)

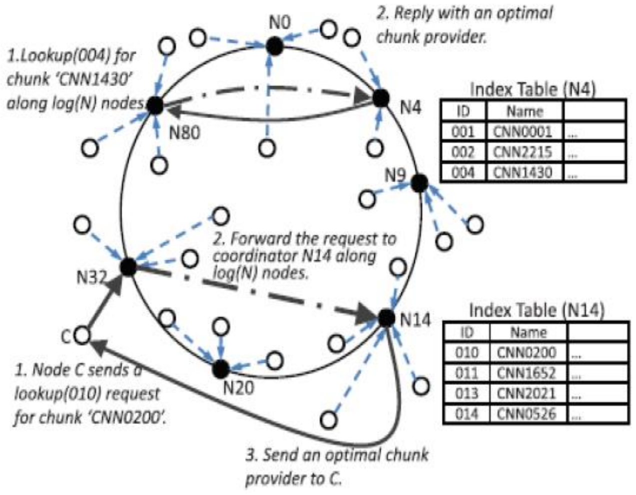

N0
2. Reply with an optimal chunk provider.
1.Lookup(004) for chunk 'CNN1430' along log(N) nodes
N4
N80
Index Table (N4)
ID | Name
001 | CNN0001 | ...
002 | CNN2215 | ...
004 | CNN1430 | ...
N9
2. Forward the request to coordinator N14 along log(N) nodes.
N32
N14
Index Table (N14)
ID | Name
010 | CNN0200 | ...
011 | CNN1652 | ...
013 | CNN2021 | ...
014 | CNN0526 | ...
C
1. Node C sends a lookup(010) request for chunk 'CNN0200'.
N20
3. Send an optimal chunk provider to C.

Figure 1: Chunk sharing in Overlay Networks

Haiying Shen et al.,[5] gives a model called Social Tube-Their work focuses on increasing scalability and user experience in online social networks for video sharing applications. They have given mainly two algorithms P2P overlay construction and chunk prefetching algorithms to address these problems. Fujimoto et al., [6] given a strategy called P2P Video-on-Demand streaming using caching and reservation scheme based on video popularity. In many algorithms a first-in, first-out approach is typically used for caching, it is not a very efficient mechanism to use peers' uploading capacity because the peers may cache the data of unpopular and not interested vide data. Their work focus on Video-Popularity-based Caching and Reservation (VPCR) scheme which makes use of the upload capacity of peers. They also address temporal viewership fluctuations for videos which reduces use of peers upload capacity using the reservation scheme VPCR. These literature surveys done here shows that each of the model given has one or other problems. By studying all these papers we incorporated DHT- Based Chunk Driven Overlay scheme which address the node failure problems in overlay construction algorithms by selecting stable nodes for chunk sharing

IV. PROPOSED METHODOLOGY

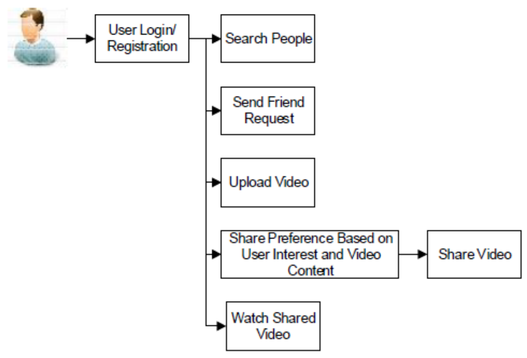


Figure 2: System Architecture

Figure 2 describes the system architecture of our proposed model. First a user can login/Register to our social networking site with his/her credentials, after registering he/she can login to the site. The successful login creates profile for a given user in our social networking site where he can find four features namely i) Search People, ii) Send Friend Request, iii) Upload Video iv) Share Video. Once a user finds people in our social networking site he can send friend requests to others to establish social connections. After successful social relation establishment he/she can upload their interested videos in the social networking site. Once the video upload is successful he/she can share the video either as public or as private by selecting few friends in which he/she can find interests

We use channel closeness of two channels to reflect the frequency of interactions between these channels, i.e., the recent tendency of nodes to switch between or watch both channels. Such a tendency can be evaluated by three factors: 1) the age (i.e., freshness) of the node's switching or multichannel watching activity on both channels; 2) the time period that the node stays in both channels; and 3) if both the channels are in the node's interested channel list.

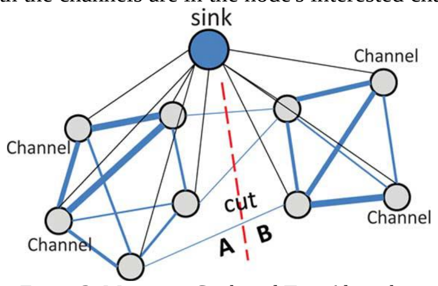


Figure 3: Minimum Cut based Tree Algorithm

Centralized Channel Clustering: Brought together Channel Clustering: In the unified bunching technique, nodes report their exercises to their channel heads, which figure channel closeness and report the data to the server. At that point, the server conducts channel grouping. The server first creates an undirected chart, where vertices speak to channels and edges speak to the communications between channels. The heaviness of the edge interfacing channels and is the entirety of channel closeness and the server then uses the base slice tree-based calculation to separate the vertices in the whole diagram to subsets, i.e., to make channel groups.

```
1 $V' = V \cup s$; //$s$ is the sink;
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 with small sum of intercluster cut
  values and large sum of intracluster cut values;
8 Return all connected subgraphs as the clusters of $G$;
```

Figure 4: Centralized Clustering Algorithm

As appeared in Fig. 3, a cut isolates all diverts in diagram into two channel subsets and . The estimation of a cut equivalents the whole of the weights of the edges crossing the cut. The base cut tree calculation makes bunches that have little total of intercluster cut qualities and moderately huge entirety of intracluster cut qualities. Calculation 1 demonstrates the pseudocode of the grouping calculation relating to the procedure in Fig. 5. To start with, we embed a simulated sink into the diagram (step 1). The sink is associated with all diverts in the chart with weight (steps 2–4). is utilized to control the quantity of produced groups. In the event that , all diverts will be in one mammoth group, while will make all channels get to be singletons. Then, we use the maximum flow algorithm [47] that involves a recursive process to construct a minimum-cut tree with the minimum sum of the values of cuts (step 5). Next, we remove the sink, and graph consequently is divided into several clusters (steps 6 and 7). The intracluster cut value can be used to measure the tightness of channels in each cluster. If a cluster has tightness higher than a predefined threshold, its channels are merged to one overlay. Otherwise, its channels build bridges between each other. The minimum cut algorithm requires a computation complexity of [8].Other clustering approaches (e.g., [9]) can also be adopted for the channel clustering in SAVE.

2) Decentralized Channel Clustering: For ease of presentation, we use *channel cluster* to denote both individual channels and a cluster of multiple channels. A cluster head is the most stable node with the highest capacity and longest lifetime staying in the cluster. The decentralized method aims to generate and maintain a stable state for the created clusters, i.e., they have small sum of intercluster closeness and relatively large sum of intracluster channel closeness.

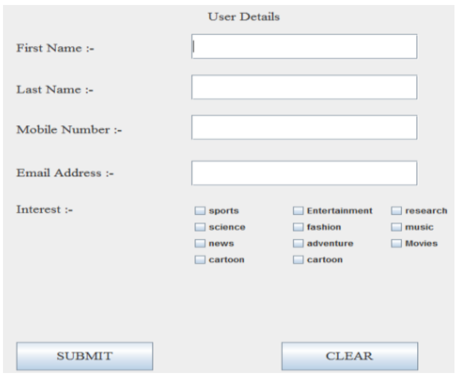
```
1  Calculate $\mathcal{V}_i$ and $S_{cr_i}$ [(2) and (3)];
2  for each interacted channel cluster $cr_k \in (\Theta - cr_i)$ do
3      $cr_{(i,k)} = cr_i \cup cr_k$;
4      Ask for information from $h_{cr_k}$;
5      for each channel cluster $cr_a \in (\Theta - cr_i - cr_k)$ do
6          Calculate $V(cr_{(i,k)}, cr_a)$ in $\mathcal{V}_{(i,k)}$ [(2)];
7      Calculate $\mathcal{P}_{(i,k)}$;
8      Calculate $S_{cr_{(i,k)}} = \mathcal{P}_{(i,k)} \cdot \mathcal{V}_{(i,k)}{}^\top$ [(3)]
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 $cr_k$
```

Figure 5: Decentralized Clustering Algorithm

Channel heads of frequently interacted channels to build or remove bridges between them. In the decentralized clustering method, the cluster heads communicate with each other for the bridge construction and elimination. In cluster combining, to build a bridge between two channel clusters (including individual channels), each channel head in one cluster builds connection with each channel head in the other cluster. The more stable and higher-capacity cluster head becomes the cluster head of the new bridged cluster. In both centralized and decentralized methods, after the bridge is built, the channel heads notify the nodes in their channels about the bridge establishment. Each channel head and nodes in a cluster maintain a record of all channels in its cluster.

VI. IMPLEMENTATION AND RESULTS

We have experimented results by creating a social networking website where we have established social connections among other users by sending and accepting friend requests. For every user who logins to our site we generated a profile. After creating social profile and establishing friendship relations. Users can upload their interested videos in our site.



User Details

First Name :-

Last Name :-

Mobile Number :-

Email Address :-

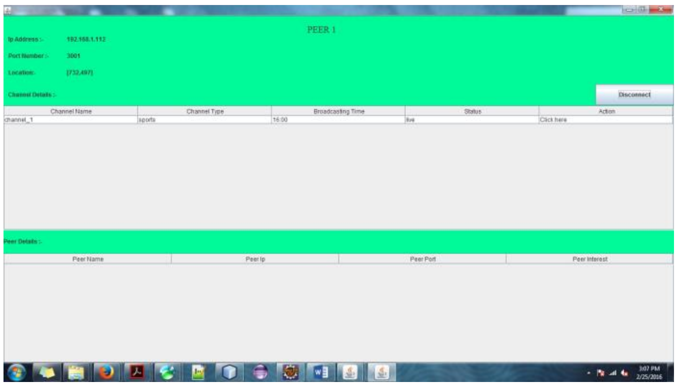
Interest :-

☐ sports	☐ Entertainment	☐ research
☐ science	☐ fashion	☐ music
☐ news	☐ adventure	☐ Movies
☐ cartoon	☐ cartoon	

SUBMIT CLEAR

Figure 6: Finding People in Online Social Networks

Later they can share videos publicly or privately among friends based on their interests. Figure shows the finding the people in social networking site with whom we can establish friendship relations by sending friend requests. It is found that friendship relations are established with the most people are tend to be located in the same place. Figure 6 shows uploading a video in social networking site. User can upload his/her interested video in the site. Most of the videos are uploaded are of short length and they were shared among friends based on their interests. Figure 6 shows a video can be shared among friends privately. They may also share publicly.



PEER 1

Ip Address :- 192.168.1.112

Port Number :- 3001

Location: [732,497]

Channel Details :-

Disconnect

Channel Name	Channel Type	Broadcasting Time	Status	Action
channel_1	sports	16:00	live	Click here

Peer Details :-

Peer Name	Peer Ip	Peer Port	Peer Interest

3:07 PM
2/25/2016

Figure 7

Fig. 8(a) and 8(b) shows the average channel switch delay versus the average node capacity (in the Pareto distribution) for the two provider selection strategies on PeerSim, respectively. From Fig. 20(a), we see that the switch delay decreases as the capacity increases for all five methods. This is because when nodes in the system have more capacity to serve chunk requests, the chunk requests are less likely to be delayed.

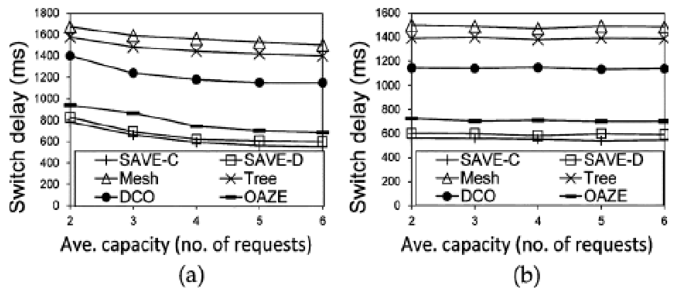


Figure 8(a), 8(b): Effectiveness of Chunk provider selection strategies comparison

VI. CONCLUSION

SAVE, an informal organization helped effective P2P live spilling framework. SAVE backings progressive and various channel seeing with low switch postpone and low server overhead by improving the operations of joining and exchanging channels. It considers the authentic channel exchanging exercises as the social connections among channels and bunches the much of the time associated channels together by consolidating overlays or assembling spans between the overlays. It expands the likelihood that current clients can find their craved channels inside its channel group and can take the extensions for channel switches. Also, every node has a friendlist that records nodes with comparative watching designs, which is utilized to join another channel overlay. SAVE additionally has the channel-closeness-based lump pushing methodology and limit based piece supplier determination technique to improve its framework execution. Our overview on client video spilling watching exercises affirms the need and achievability of SAVE. we demonstrate that SAVE outflanks other agent frameworks as far as overhead, video gushing proficiency and server load lessening, and the adequacy of SAVE's two systems. Our future work lies in further decreasing the expense of SAVE in structure upkeep and node correspondence.

REFERENCES

[1] Social Media and http://thefuturebuzz.com/

[2] J. Liu andX. Cheng. NetTube: Exploring Social Networks for Peer-to- Peer Video Sharing. In Proceeding of INFOCOM, 2009.

[3] C. Huang, J. Li, and K. W. Ross. Can Internet Video on-Demand be Profitable? In Proceedings of SIGCOMM, 2007.

[4] H. Shen, L. Zhao, Z. Li, and J. Li. A DHT-aided Chunk-driven Overlay for Scalable and Efficient P2P Live Streaming. IEEE Transactions Vol. 24 No. 11.Pages 2125-2137, 2013.

[5] Z. Li, H. Shen, H.Wang, G. Liu, and J. Li. SocialTube: P2P Assisted Video Sharing in Online Social Networks. Vol. PP No. 99 of IEEE Transactions, 2014.

[6] T. Fujimoto, R. Endo, and H. Shigeno. P2P Video-on Demand Streaming Using Caching and Reservation Scheme based on Video Popularity (VPCR). International Journal of Grid and Utility Computing, 3(2/3):188–199, 2012.

[7] W. P. K. Yiu, X. Jin, and S. H. G. Chan. VMesh: Distributed Segment Storage for Peer-to-Peer Interactive Video Streaming. IEEE Journal on Selected Areas in Communication, 2007.

[8] H. Shen, L. Zhao, H. Chandler, J. Stokes, and J. Li. P2P Based Multimedia Sharing in User Generated Contents. IEEE Transactions on Parallel and Distributed Systems, Volume. 23, No. 5, May 2012.

[9] Haiying Shen, Yuhua Lin, and Jin Li, Social Network-Aided Efficient Peer-to-Peer Live Streaming System. IEEE Transctions. Vol.23, 2015.

[10] A. Fast, D. Jensen, and B. Levine, "Creating social networks to improve peer-to-peer networking," in *Proc. ACM SIGKDD*, 2005, pp. 568–573.

[11] A. Iamnitchi, M. Ripeanu, and I. Foster, "Small-world file-sharing communities," in *Proc. IEEE INFOCOM*, 2004, pp. 952–963.

[12] J. Kleinberg, "The small-world phenomenon: An algorithmic perspective," in *Proc. STOC*, 2000, pp. 163–170.

[13] S. Milgram, "The small world problem," *Psychol. Today*, vol. 2, no. 1, pp. 60–67, 1967.

[14] I. Bermudez, M. Mellia, and M. Meo, "Investigating overlay topologies and dynamics of P2P-TV systems: The case of SopCast," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 9, pp. 1863–1871, Oct. 2011.

[15] F. Wang, J. Liu, and Y. Xiong, "Stable peers: existence, importance, and application in peer-to-peer live video streaming," in *Proc. IEEE INFOCOM*, 2008, pp. 2038–2046.

[16] J. Mendes, P. Salvador, and A. Nogueira, "P2P-TV service and user characterization," in *Proc. IEEE CIT*, 2010, pp. 2612–2620.

[17] "The PeerSim simulator," 2013 [Online]. Available: http://peersim.sf.net

[18] "PlanetLab," [Online]. Available: http://www.planet-lab.org/

[19] 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.

[20] 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.

[21] Y. Chu, A. Ganjam, T. Ng, S. Rao, K. Sripanidkulchai, J. Zhang, and H. Zhang, "Early experience with an internet broadcast system based on overlay multicast," in *Proc. USENIX*, 2004, p. 12

[22] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proc. SIGCOMM*, 2002, pp. 205–217.

[23] Y. Chu, S. Rao, and H. Zhang, "A case for end system multicast," in *Proc. ACM SIGMETRICS*, 2000, pp. 1–12.

[24] R. Tian, Q. Zhang, Z. Xiang, Y. Xiong, X. Li, and W. Zhu, "Robust and efficient path diversity in application-layer multicast for video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 8,pp. 961–972, Aug. 2005