SocialTube: P2P-assisted Video Sharing in Online Social Networks

Ze Li, Haiying Shen, Hailang Wang and Guoxin Liu
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29631
Email: {zel, shenh, hailang, guoxinl}@clemson.edu

Jin Li
Microsoft Research
Redmond, WA 98052
Email: jinl@microsoft.com

***Abstract*—Video sharing has been an increasingly popular application in online social networks (OSNs). However, its sustainable development is severely hindered by the intrinsic limit of the client/server architecture deployed in current OSN video systems, which is not only costly in terms of server bandwidth and storage but also not scalable. The peer-assisted Video-on-Demand (VOD) technique, in which participating peers assist the server in delivering video content has been proposed recently. Unfortunately, videos can only be disseminated through friends in OSNs. Therefore, current VOD works that explore clustering nodes with similar interests or close location for high performance are suboptimal, if not entirely inapplicable, in OSNs. Based on our long-term real-world measurement of over 1,000,000 users and 2,500 videos in Facebook, we propose SocialTube, a novel peer-assisted video sharing system that explores social relationship, interest similarity, and physical location between peers in OSNs. Specifically, SocialTube incorporates three algorithms: a social network (SN)-based P2P overlay construction algorithm, a SN-based chunk prefetch algorithm, and a buffer management algorithm. The trace driven based simulation results show that SocialTube can improve the quality of user experience and system scalability over current P2P VOD techniques.**

I. Introduction

Video sharing has been an increasingly popular application in Online Social Networks (OSNs) (e.g., Facebook [1], Twitter [2]). According to comScore Releases in August 2010, Facebook is now the second-largest online video viewing platform. The total time spent on video viewing on Facebook increased 1,840% year-over-year, from 34.9 million minutes in October 2008 to 677.0 million minutes in October 2009. However, OSN's further advancement is severely hindered by the intrinsic limits of the conventional client/server architecture of its video sharing system, which is not only costly in terms of server storage and bandwidth but also not scalable with the soaring number of users and video content in OSNs.

P2P-based video sharing has been used in on-demand video streaming (e.g., GridCast and Vanderbilt VoD). With each peer contributing its bandwidth to serving others, the P2P architecture provides high scalability for large user bases. Previous P2P VoD systems either randomly cluster peers for video inquiry [3]–[6] or form certain peers into a distributed hash table (DHT) for chunk indexing [7]–[9]. In order to reduce the video transmission and/or prefetching delay, some works cluster nodes with close physical proximity [7], [10], [11] or similar interests [9], [12]. However, those mechanisms are suboptimal, if not entirely inapplicable, in OSNs. Unlike VoD systems that provide system-wide video searching and sharing, where a peer can access any other peer's content, OSNs do not provide video search functionality. In an OSN, videos are visited and spread by the users' friends through the Friend-of-Friend (FOF) relationship. Therefore, users in an OSN watch videos driven by both the friendship relation and video content.

In order to investigate the video watching behaviors of users in OSNs, we crawled data from more than 1,000,000 users and 2,500 videos in Facebook. Our measurement reveals that (1) most of the viewers of a user's videos are the user's close friends and (2) most video views are driven by social relationships, and the rest are driven by interests. Based on our observations, we propose SocialTube, a system that explores the social relationship and interest similarity to enhance the performance of video sharing in OSNs. Specifically, SocialTube has a social network (SN)-based P2P overlay construction algorithm that clusters peers based on their social relationships and interests. SocialTube also incorporates an SN-based video prefetching algorithm to increase the video prefecth accuracy to minimize video playback startup delay. To our knowledge, this work is the first that studies the distinct characteristics of OSN video sharing and builds a P2P-based video sharing system in an OSN by leveraging those characteristics.

II. Facebook Measurement and Analysis

In this section, we present our Facebook trace measurement results and give an in-depth perspective of Facebook video viewing patterns. We used breadth-first-search [13] to crawl data from over 1,000,000 users seeded by 5 users in the USA. Because of the privacy constrain and the fact that many users do not have videos, we only found 2,500 videos and about 12,000 users who watched these videos, which is used as a sample of the video sharing and watching activities between Jul. 2007 to Aug. 2010. The collected dataset includes information about user friendship relations, interests, location, and videos uploaded and shared by users. For each video, we retrieved its title, length, and viewers when available.

A. *Effect of Social Distance on Video Viewing Patterns*

At first, we investigate the impact of social distance on user video viewing patterns. Among 52,500 video watching activities involving 12,000 users, we measured the social distance of

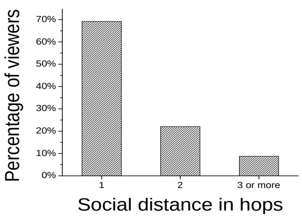

Fig. 1: Social distance.

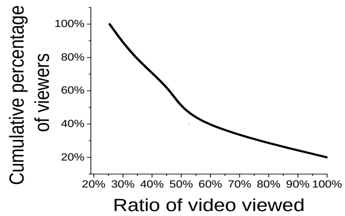

Fig. 2: Different viewer types.

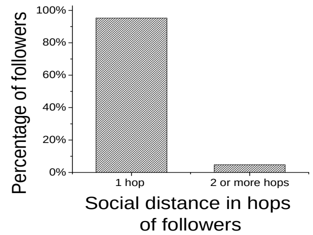

Fig. 3: Social distance between follower and video owners.

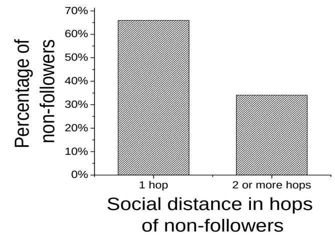

Fig. 4: Social distance between non-follower and video owners.

a video viewer from the video owner, and show the distribution in Figure 1. We find that most of the viewers (around 70%) are 1-hop friends of the video owner, 2-hop viewers account for a portion of about 20%, and the remaining 10% of viewers watched videos of video owners are more than 2 hops away.

O1: *In Facebook, more than 90% of the viewers of a video are within 2 hops in the video owner's social network.*

We define a video viewer group as all users who have watched the video owner's videos. From O1, we obtain the inference (I):

I1: *A video viewer group of a video owner in Facebook is mostly within the 2-hop friend circle of the owner.*

Note that a user may own more than one video. To further identify the impact of social relationships on video viewing patterns, we selected the users who have multiple videos from our dataset and inspected the viewer group of each video owner. We classified the viewers of a video owner based on the ratio of videos of all videos from the owner they watched. and calculated the percent of different viewer categories in a viewer group. Figure 2 shows the average values of the percent versus the ratio of videos watched from the video owners.

O2: *On average, in a user's viewer group, 25% of viewers watched all, 33% of viewers watched 80%, and all viewers watched 20% of the user's videos.*

We call the viewers who have watched almost all videos of a user the user's *followers*, and call other viewers *non-followers*. We use a threshold T_h for the percent of all the videos of a user that a viewer watches in order to become a follower, and set T_h=80% in this analysis. Figure 3 and Figure 4 show the distribution of followers and non-followers in terms of the social distance with the video owner. :

O3: *Viewers that watch almost all of a user's videos (i.e., followers) usually are 1-hop friends of the user, while most of other viewers (i.e., non-followers) are 1-hop or 2-hop friends of the user.*

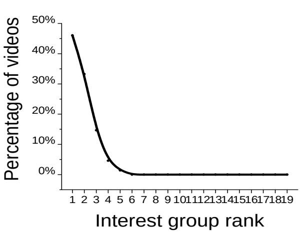

Fig. 5: Interest clustering effect

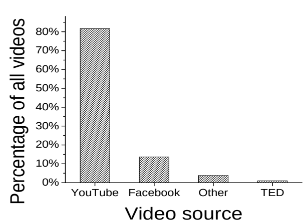

Fig. 6: Video source in Facebook.

B. Effect of Interest on Video Viewing Pattern

Next, we explore the correlation between user interests and video viewing patterns. We select a sample of 118 distinct users that watched more than one video from our dataset and manually classify the videos they watched into 19 interest groups based on video content. The 19 interest groups were determined based on the video categories in YouTube such as gaming, rock music and action movie. For each user, we calculated the percentage of viewed videos of each interest group. Then, we ranked these 19 interest groups in descending order of the percentage values. We calculated the average percentage value of the 118 users for each interest group rank and show the result in Figure 5. We observe that, on average, 46% of videos a user watched are on his/her top 2 interests topics, 79% of videos a user watched are on his/her top 3 interests topics, and 94% are on his/her top 4 interests topics. The result implies that the videos each user watches are generally orientated towards his/her few primary interests.

O4: *Users tend to watch the videos of their interests and each user generally has* ≤ 4 *video interests.*

A user can post on Facebook either self-uploaded videos or external video links from a third party video service provider such as YouTube. The video linking in Facebook is called "share", by which users can share links to videos they find interesting with their friends. Figure 6 shows the different sources of videos in our collected dataset. We see that the self-uploaded videos in Facebook account for about 14% of all videos. Others are external video links. YouTube, as the largest video site in the world, accounts for over 80% of all external links. Many other video sources such as TED and Hulu account for the remaining percentage.

O5: *A large percentage of videos in Facebook are from YouTube, where the user video viewing patterns are driven by interests.*

Combining O1-O5, we can find that different watching incentives can be applied to different types of viewers. The followers of a user watch most of the user's videos regardless of the video content because of their close social relationship (e.g., close friends and fervent admirers). For the viewers that watch only a few of the user's videos, interest in the video content is a more important incentive. Additionally, some show a mixed video watching incentive. Thus, we infer

I2: *Followers are primarily driven by social relationship to watch videos, while non-followers are driven mainly by interest.*

III. THE DESIGN OF SOCIALTUBE

Based on observations O1 - O5, we propose SocialTube, a P2P video sharing system for OSNs. In this paper, we use *server* to represent all video source servers, including both Facebook and external video servers. Similar to current peer-assisted content delivery mechanisms, the peers in SocialTube

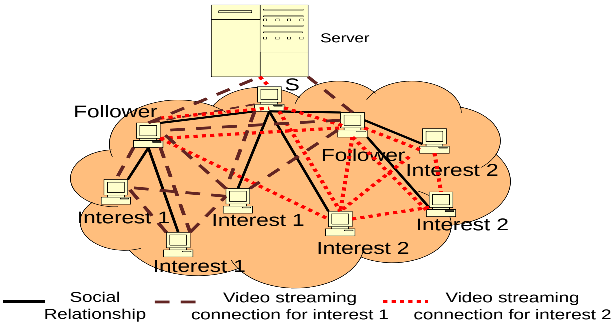

Fig. 7: Structure of SocialTube.

store videos they have watched for video re-distribution. In SocialTube, a video is divided into small chunks with a fixed size. Thus, a watching user only needs to download the corresponding chunks of the video segment to watch.

A. Social Network based P2P Overlay Construction Algorithm

To identify followers and non-followers of a source node for structure construction, SocialTube pre-defines two thresholds, T_h and T_l, for the percent of videos in the source node that a viewer has watched during a time unit, say one week. If the percent value of a viewer is $\geq T_h$, the viewer is a follower. If the percent is $T_l < x < T_h$, the viewer is a non-follower.

Figure 7 shows a basic P2P overlay structure. Based on I1, SocialTube establishes a *per-node* (in contrast to *per-video* in YouTube) P2P overlay for each source node, which consists of peers within 2 hops to the source that watch at least a certain percentage ($> T_l$) of the source's videos. Other peers can still fetch videos from the server. As shown in the figure, such peers of S in the social network constitute a P2P overlay for a source node "S". Based on I2, we build a hierarchical structure that connects a source node with its socially-close followers, and connects the followers with other non-followers. Thus, the followers can quickly receive chunks from the source node, and also function as a pseudo-source to distribute chunks to other friends. The source pushes the first chunk of its new video to its followers. The chunk is cached in each follower and has high probability of being used since followers watch almost all videos of the source. Further, non-followers sharing the same interest are grouped into an interest cluster for video sharing. We call peers in an interest cluster *interest-cluster-peers*. A node that has multiple interests is in multiple interest clusters of the source node. Because the source node and followers are involved in every interest cluster for providing video content, we call the group formed by the source, followers, and interest-cluster-peers in an interest cluster *swarm*, and call all nodes in a swarm *swarm-peers*. As I1 indicates, the cluster size of each interest cluster should be small. In Figure 7, the viewers of S form into two swarms.

In current video sharing in Facebook, a node always requests the server for videos uploaded by source nodes. We let the server keep track of the video watching activities of viewers of a specific source node in order to identify and update its followers and non-followers based on SocialTube's pre-defined thresholds of T_l and T_h. This duty can be assigned to the source node itself if it has sufficient capacity. The nodes in the system will periodically report their video watching activities to the server. When the server determines that a peer is a follower of the source node, it notifies the source node, which notifies all nodes in its swarms about the follower. Consequently, the follower becomes a member of each of the swarms, and all swarm-peers in each of the swarms connect to it. If the peer is a non-follower, the server determines its interests based on the video labels the peer visited, and notifies the source node about the non-follower along with its interests. The source node then notifies the peers in the clusters of the interests of that non-follower, and notifies the non-follower about the clusters. The non-follower connects to all followers and the source and to a number of nodes in each cluster. Consequently, the non-follower becomes a member of the swarm of each of the interest clusters. The server periodically updates the roles of the followers and non-followers. The nodes in a P2P structure, including the source, followers and non-followers, remember their roles and connections. Next time when a node goes online, it automatically connects to its previous neighbors and function based on its role. As indicated in Figure 7, the source node has two followers, and its videos can be divided into two interest categories based on video content. The 1-hop and 2-hop friends of the source node with interest 1 and interest 2 form into two clusters, respectively. The source node and the followers are in each interest cluster, all of which form a swarm.

B. Social Network based Prefetching Algorithm

To reduce the video startup latency, we propose a push-based video prefetching mechanism in SocialTube. In SocialTube, when a source node uploads a new video to the server, it also pushes the prefix (i.e. first chunk) of the video to its followers and to the interest-cluster-peers in the interest clusters matching the content of the video. The prefix receivers store the prefix in their cache. Those interest-cluster-peers and followers who are not online when the source node pushes the prefix will automatically receive it from the source node or the server once they come online. After the source node leaves, the responsibility to push the prefix falls to the server. Since these followers and interest-cluster-peers are very likely to watch the video, the cached prefixes have a high probability of being used.

Once the nodes request the videos, the locally stored prefix can be played immediately without delay. Meanwhile, the node tries to retrieve the remaining video chunks from its swarm-peers. Similar to BitTorrent, SocialTube allows a requester to request 4 online nodes at the same time to provide the video content in order to guarantee provider availability and achieve low delay by retrieving chunks in parallel. It first contacts interest-cluster-peers, then followers, then the source node. If the requester still cannot find 4 providers after the source node is contacted, it resorts to the server as the only provider. Considering the high capacity of the server, the requester does not need to have 4 providers if it has the server as a provider. This querying order can distribute the load of chunk delivery among the swarm-peers while providing high chunk availability. The algorithm takes advantage of all resources for

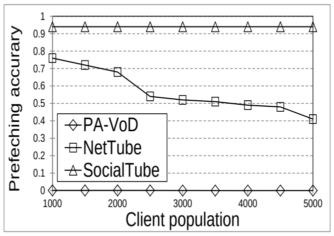

Fig. 8: Prefetching accuracy vs. node population.

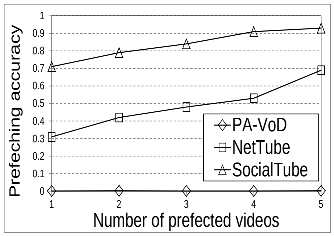

Fig. 9: Prefetching accuracy vs. # of prefetched videos.

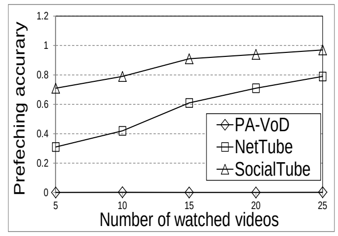

Fig. 10: Prefetching accuracy vs. # of watched videos.

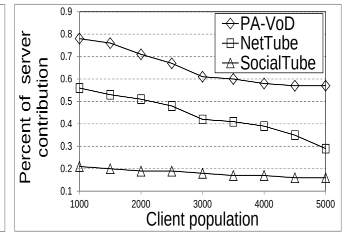

Fig. 11: Percent of server contribution vs. client population.

efficient video sharing without overloading specific nodes. The server can guarantee the availability of the video, even if the number of online users in a swarm is small.

IV. Performance Evaluation

A. Experiment Settings

Our simulation setup is based on a part of our crawled dataset (Sep. 2010). In order to create a network that conforms to a power law distribution [14], we selected 5,000 out of 12,000 nodes out of the trace data, which contains approximately 2,000 related and distinct videos. We assign each of the 2,000 videos to a randomly chosen node from the 5,000 nodes. We did not directly use the video ownership in the trace is because some video owners are not in 5,000 nodes.

To simulate the real Internet environment, where the peers are heterogenous in terms of bandwidth, we use the bandwidth statistics used in [12]. Table 1 shows the default parameters, unless otherwise specified. The video bitrate is set to 330 kbps. Based on empirical observations, we assume that, whenever a buffer underflow occurs, the peer pauses for 3 seconds and then resumes playback. The average file size of a video is randomly chosen from [20-30] MB, which is normal for a video with a length of 2-3 minutes.

TABLE I: Experiment default parameters.

Parameter	Default value
Number of clients	5,000
Number of videos	2,000
Number of interest categories	19
Number of interests per client	2-4
Trace duration	40 days
Chunk size	3 MBytes
Prefix length	3 MBytes
Server uploading bandwidth	20 Mbps
Video size	Distribution of YouTube videos
Cache szie	300M

The simulation is conducted based on simulation cycles. One simulation lasts for 100 simulation cycles. Each simulation cycle is used to simulate one day. Since every person spends about 420 minutes on Facebook per month [15] on average, we can infer that every person spends about 14 minutes and watches at most 4 videos per day on average. Therefore, in the experiment, the number of videos a peer watches in each simulation cycle is randomly chosen from the range [1, 4]. Based on O4, each node randomly selects [2, 4] interest categories as its interests and only watches the videos in its interests.

Based on O3, every source node randomly selects 25% of its friends as its followers. To simulate user video viewing behaviors in Facebook, every node knows the IDs of the videos in its 1-hop friends. In each simulation cycle, a source node only pushes one video prefix to its followers and the cluster-peers in the interest cluster matching the video. Recall that due to privacy protection in OSNs, a node's videos cannot be accessed by others with more than 1 hop social distance unless the videos are shared by its friends. To select videos to share, a node first randomly selects a number $x \in [1, 40]$. After it watches the x^{th} video, it shares the video with its friends and then randomly selects another number $y \in [1, 40]$. Then, after the node watches the $(x+y)^{th}$ video, it shares the video with its friends. This process is repeated until the simulation completes. To simulate the geographic locations of nodes, each node has a location ID in [1, 10]. The nodes with numerically closer IDs are closer nodes.

We compare the performance of SocialTube with two other representative works in peer-assisted video streaming, PA-VoD [11] and NetTube [12]. In PA-VoD, physically close peers with the same location ID are clustered for video sharing between each other. In NetTube, peers that have similar interests are clustered together for video sharing. In order to reduce the startup time of NetTube and PA-VoD, we let the server push the prefix of each video to the nodes in the interest cluster of the video in NetTube, and to the nodes physically close to the video's uploader in PA-VoD. To watch a video in an interest, a node searches videos in its cluster. In PA-VoD, a requester first looks for a video in its cluster. If there are no videos within the requester's interests owned or shared by its friends in the searched cluster, the cluster which is 2 hop away from the requester is searched, and so on. If a node cannot find a requested video from the peers, it resorts to the server. In SocialTube, a node only searches one cluster in its interest though it may be in multiple clusters. We focus on following two metrics in the experiments: (1) Prefetching accuracy. This is the probability that a user requests a video whose prefix is in its cache. (2) Percent of server contribution. This is the ratio of server bandwidth consumed in SocialTube over the total bandwidth consumed in the client/server system.

B. Effectiveness of the Prefix Prefetching Mechanism

One goal of SocialTube is to reduce video playback startup delay. SocialTube uses a push-based prefix prefetching mechanism in order to reduce the user waiting time for video startup. Figure 8 shows the prefetching accuracy versus client (i.e., node) population, which is varied from 1,000 to 5,000. The figure indicates that as the client population increases, the node prefetching accuracy of SocialTube and PA-VoD remains almost constant, while that of NetTube decreases significantly. NetTube only clusters peers with similar interests without considering the social relationships between peers. When the server pushes the video prefixes to these nodes with similar interests to the video, some pushed prefixes are unused because peers only issue video requests to their friends. As the client population increases, more requests from the clients cannot be resolved by the prefix, leading to a decreasing prefetching accuracy. In contrast, SocialTube explores the social relationship between peers and clusters peers with similar interests together. Therefore, the video prefixes are sent to socially-close nodes that are able to watch the video, leading to higher prefetching accuracy. As the client population increases, the size of the interest clusters in SocialTube may increase. Since most cluster-peers within 2-hops of the video owner are able to visit each other, the video prefetching accuracy remains nearly constant. However, in PA-VoD, the nodes are clustered based on physical location rather than interests. Thus, nodes in a cluster that receive video prefix are unlikely to be interested in the video as they may not be friends or share same interests.

Figure 9 shows the prefetching accuracy versus the number of prefetched videos of a node. In the experiment, we randomly selected 5 nodes and monitored the number of video prefixes they receive and their corresponding prefetching accuracy and reports the average value of the prefetching accuracies of the five nodes. The figure shows that as the number of prefetched videos increases, the prefetching accuracy of SocialTube and NetTube increases. This is because as a peer receives more video prefixes, it has a higher probability of having the requested prefix in its cache. NetTube leads to a lower prefetching accuracy than SocialTube for the same reason as in Figure 8. We also find that the prefetching accuracy in PA-VoD stays around 0. Although a client in PA-VoD caches more prefixes, these prefixes may not match any of its interests, which leads to a small hit rate.

We show the prefetching accuracy versus the number of a node's watched videos in Figure 10. As in Figure 9, we randomly selected 5 nodes and monitored the number of videos they watched. The figure illustrates that as the number of watched videos increases, the prefetching accuracy of the protocols increases. In SocialTube and NetTube, as a node watches more videos, the interest clustering accuracy improves. Therefore, the prefixes are more accurately pushed to the nodes that are likely to watch the video. Again, NetTube generates a lower prefetching accuracy than SocialTube for the same reason as in Figure 8. In PA-VoD, since the nodes are not clustered based on their interests, an increased number of watched videos does not affect the hit rate. Therefore, the prefetching accuracy of PA-VoD still remains low.

C. Contribution of Servers

Figure 11 illustrates the percent of server contribution versus client population. It shows that as the client population increases, the percentage of server contribution in all of PA-VoD, NetTube and SocialTube decreases. As more nodes join in the system, more bandwidth is contributed for P2P video transmission from peers, thus reducing the bandwidth consumption of the server. We can also see that SocialTube contributes much more P2P bandwidth than NetTube and PA-VoD. This is because the nodes in SocialTube can locate the video chunks from other peers more efficiently than in PA-VoD and NetTube, since SocialTube considers both social relationship and interest. As the nodes in NetTube can locate chunks more efficiently than those in PA-VoD because NetTube considers interest, the peer contribution of NetTube is higher than PA-VoD.

V. Conclusion

The client/server architecture deployed by current video sharing systems in OSNs costs a large amount of resources for the service provider and lacks scalability. Meanwhile, because of the privacy constraints in OSNs, the current peer-assisted Video-on-Demand (VoD) techniques are suboptimal if not entirely applicable to the video sharing in OSNs. In this paper, we presented the video watching trace data in one of the largest online social network websites Facebook, from Jul. 2007 to Aug. 2010 and explored the users' video viewing patterns. We found that in a user's viewer group, 25% viewers watched all videos of the user driven by social relationship, and the viewing pattern of the remaining nodes is driven by interest. Based on the observed social and interest relationship in video watching activities, we propose SocialTube, which provides efficient P2P-assisted video sharing services. Numerous simulation results show that SocialTube can provide a high video prefetch accuracy and low server traffic demand.

Acknowledgements

This research was supported in part by U.S. NSF grants NSF-CSR 1025649, OCI-1064230, CNS-1049947, CNS-1156875, CNS-0917056 and CNS-1057530, CNS-1025652, CNS-0938189, Microsoft Research Faculty Fellowship 8300751, and Sandia National Laboratories grant 10002282.

References

[1] Facebook. http://www.facebook.com/.
[2] Twitter. http://twitter.com/.
[3] K. Wang and C. Lin. Insight into the P2P-VoD system: Performance modeling and analysis. In *Proc. of ICCCN*, 2009.
[4] Y. Huang, Z. Fu, D. Chiu, C. Lui, and C. Huang. Challenges, design and analysis of a large-scale P2P VoD system. In *Proc. SIGCOMM*, 2008.
[5] B. Cheng, L. Stein, H. Jin, X. Liao, and Z. Zhang. Gridcast: improving peer sharing for p2p vod. *ACM TOMCCAP*, 2008.
[6] Chien-Peng Ho, Suh-Yin Lee, and Jen-Yu Yu. Cluster-based replication for P2P-based video-on-demand service. In *Proc. of ICEIE*, 2010.
[7] W. P. K. Yiu, X. Jin, and S. H. G. Chan. VMesh: Distributed Segment Storage for Peer-to-peer Interactive Video Streaming. *IEEE JSAC*, 2007.
[8] H. Shen, L. Zhao, Z. Li, and J. Li. A DHT-aided chunk-driven overlay for scalable and efficient P2P live streaming. In *Proc. of ICPP*, 2010.
[9] H. Shen, L. Zhao, H. Chandler, J. Stokes, and J. Li. P2P-based multimedia sharing in user generated contents. In *Proc. of INFOCOM*, 2011.
[10] J. Wang, C. Huang, and J. Li. On ISP-friendly rate allocation for peer-assisted VoD. In *Proc. of MM*, 2008.
[11] C. Huang, J. Li, and K. W. Ross. Can internet video-on-demand be profitable? In *Proc. of SIGCOMM*, 2007.
[12] X. Cheng and J. Liu. NetTube: Exploring social networks for peer-to-peer short video sharing. In *Proc. of INFOCOM*, 2009.
[13] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou. Walking in facebook: A case study of unbiased sampling of osns. In *Proc. of INFOCOM*, 2010.
[14] A. Nazir, S. Raza, and C. Chuah. Unveiling dacebook: a measurement study of social network based applications. In *Proc. of SIGCOMM*, 2008.
[15] Facebook users average 7 hrs a month in january as digital universe expands. http://blog.nielsen.com/nielsenwire.