FaceTube: Efficient Video on Demand Technique in Online Social Networks

James Karunakar G[1], Ganesh D[2]
[1]M. Tech, Dept. of CSE
Sree Vidyanikethan Engineering College
Tirupati-517 102, Andhra Pradesh, India

[2]Assistant Professor, Dept. of CSE
Sree Vidyanikethan Engineering College
Tirupati-517 102, Andhra Pradesh, India

Abstract
Video sharing has gained popularity as an application in online social networks (OSNs). But, because of the limitations posed by the client/server architecture present in the OSN Video systems, its development has been impeded. The main limitations are the costs of server bandwidth, storage and the increasing number of online users and video content. The peer-assisted Video-on-Demand (VoD) technique, has recently been proposed, through which, online peers help the server in the obtaining and delivery of online content. However, videos can only be shared with friends in OSNs. Hence, present VoD works that use the technique of clustering nodes with similar interests or nearby location for better performance are not optimal, if not entirely inapplicable, in OSNs. Based on a real-world measurement over a long period of time of nearly 1,000,000 users and 2,500 videos on Facebook, we propose FaceTube, a new peer-assisted video sharing system that researches social relationship, similarity in interest, and physical proximity between different peers in OSNs. Particularly, FaceTube uses four algorithms: an online social network (OSN)-based Peer-to-peer overlay construction algorithm, an OSN-based chunk pre fetching algorithm, chunk scheduling and delivery algorithm, and an algorithm for buffer management. Experimental results from a prototype on PlanetLab and simulation results show that FaceTube would improve the quality of user experience and system scalability over current P2P VoD techniques.

Key words: On-line social networks, Video-on-demand (VoD), Peer-to-peer assisted VoD, Peer-to-peer networks by commas.

1. Introduction

Online social networks (OSNs) (e.g., Twitter, Facebook) are now the most popular sites on the Web. An OSN is a powerful means of providing social connections and finding content, organizing and sharing. For example, Facebook has over 1.32 billion monthly users. OSNs are formed around users, different from the present file or video sharing systems (e.g., BitTorrent and YouTube), which are mainly organized around content. OSN users add real world or virtual friends, update their profiles and post content like photos, videos, and notes. Video sharing has become a highly popular application in OSNs, enabling users to share interesting videos of them or the ones found elsewhere on the internet. According to comScore Releases in August 2010, Facebook has become the second-largest online video viewing platform. The total amount of time spent on video viewing on Facebook has increased exponentially, at 1,840% year-over-year, from 34.9 million minutes in October 2008 to 677.0 million minutes in October 2009. During the same time frame, the number of unique video viewers increased by 548% and total number of videos grew by 987% [1]. OSNs have evolved from being simple communication focused tools to a full-fledged media portal. They are slowly transforming from being a platform for catching up with friends to a platform for personal expression and for sharing content and information But, OSN's further development is severely impeded by the limits of the conventional client/server architecture of its video sharing system, which is costly in terms of server storage and bandwidth and not scalable with the increasing amount of users and video content in OSNs. For example, YouTube, the world's largest video sharing website, spends more than $1,000,000/day for its server bandwidth. This was one of the major reason that YouTube was sold out to Google. Facebook is now facing the same problem as YouTube, as more and more users rely on Facebook for video sharing. Though many OSNs depend on content delivery networks (CDNs) for the delivery of video content(e.g., Akamai handles video delivery on Facebook), the CDN service is costly. In recent times, a lot of effort has been spent on improving the client/server architecture for video sharing, with the peer-to-peer (P2P) architecture being the most promising. P2P-based online video sharing has previously been used in on demand online video streaming (e.g., GridCast and Vanderbilt VoD). The P2P architecture can provide high scalability for large user bases, as each peer contributes its bandwith to serving others. Previous P2P VoD systems either used random clustering for video inquiry or a distributed hash table (DHT) for chunk indexing. Some works, in the past, have clustered nodes with similar interests or close physical proximity to reduce the prefetching and/or video transmission delay. However, these mechanisms are not optimal, and are not entirely applicable in OSNs. VoD systems provide system-wide video searching and sharing, where a peer can access any other peer's content, whereas, OSNs do not provide video search functionality. Videos are shared by user's friends in Facebook through the Friend-of-Friend (FOF) relationship. Particularly, if a friend of user A, say user B, receives notification when user A uploads a video, and user B's friend C is informed about the video only when user B "shares" the video, and so on. Facebook has different options to share videos only between friends, FOFs (default), or everyone. As a result, Faceboook users watch videos driven much more by the friendship, than the video content.

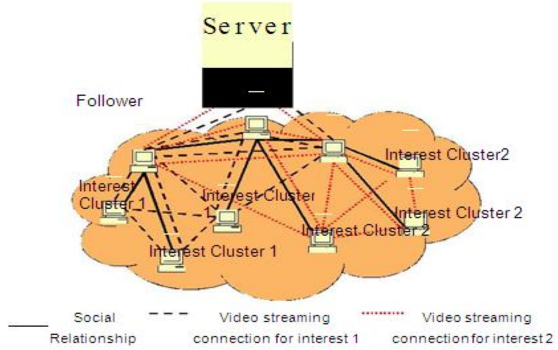

Fig. 1: Structure of FaceTube

2. Related Work

Peer-to-peer video-on-demand (P2P-VoD) is one of the most commercially important and technically challenging applications. In such systems, each video file is typically divided into sub-pieces that are stored possibly at different sending peers, and the receiving peer often needs to coordinate multiple senders by specifying a transmission schedule for them. In [2], the authors investigate the transmission scheduling problem for a specific P2P-VoD system, which is PPLive. Based on the delay analysis, an optimization problem is formulated to design the schedule for sending peers to transmit their sub-pieces. The objective of this optimization is to maximize a defined reward and at the same time meet the delay constraint required to ensure video quality at the receiver. Unlike streaming live content, P2P-VoD has less synchrony in the users sharing video content, therefore it is much more difficult to alleviate the server loading and at the same time maintaining the streaming performance. To compensate, a small storage is contributed by every peer, and new mechanisms for coordinating content replication, content discovery, and peer scheduling are carefully designed. In [3], the authors describe and discusse the challenges and the architectural design issues of a large-scale P2P-VoD system based on the experiences of a real system deployed by PPLive. After analyzing a large amount of collected data, number of results on user behavior, various system performance metrics, including user satisfaction is presented. The recent three years have witnessed an explosion of networked video sharing, represented by YouTube, as anew killer Internet application. Their sustainable development however is severely hindered by the intrinsic limit of their client/server architecture. Long-term measurement of over five million YouTube videos, on the other hand, reveal interesting social networks with strong clustering among the videos, thus opening new opportunities to explore. In [4], the authors propose NetTube, a novel peer-to –peer assisted delivering framework that explores the clustering in social networks for short video sharing is proposed. In [5], the authors propose SocialTube, a novel peer-assisted video sharing system that explores social relationship, interest similarity, and physical location between peers in OSNs.

3. Proposed algorithm

We propose two Algorithms:

1. Online social network (OSN)-based Peer-to-peer overlay construction algorithm
2. Online Social Network based Pre-fetching Algorithm

Design Considerations:

All the nodes considered in the algorithm are active Facebook users who view videos

Description of the Proposed Algorithms:

1. Online social network (OSN)-based Peer-to-peer overlay construction algorithm

Aim of the proposed algorithm is to cluster peers based on their social relationships and interests.

Step 1

Define Thresholds Define two thresholds, Th and Tl, for the percentage of videos in that a viewer has watched during a time unit. If the percentage value of a viewer is $\geq$ Th, then the viewer is said to be a follower. If the percent is Tl $< x \leq$ Th, the viewer is said to be a non-follower. Establish a per-node (as opposed to per-video in YouTube) P2P overlay for each source node. It contains of friends within 2 hops to the source node that watch at least a certain percentage (> Tl) of the source node's videos. Such peers of a source node (source friend) S in the online social network constitute a P2P overlay for the source node.

Step 2

Build a hierarchical structure The second step is to build a hierarchical structure that connects a source node with its socially-close followers, which, in turn, connects the followers with other non-followers. The source node pushes the first chunk of its new video to its follower nodes. The chunk is then cached in each follower and has a high probability of being used, since follower nodes watch almost all videos of the source node. Also, non-followers having the same interest are grouped into another cluster called interest cluster for video sharing. Peers grouped in an interest cluster are termed interest-cluster-peers. Since the source node and followers are involved in every interest cluster for providing video content, the group formed by the 'source nodes, follower nodes, and interest cluster-peers' is called an 'Interest cluster swarm'. All nodes in a swarm all called swarm-peers.

Step 3

Random connection of peers Most viewers of a video are usually physically close peers. Hence, in order to reduce transmission delay, physically proximate interest-cluster-peers are randomly connected with each other. Based on their Internet Protocol Address, subnet information, the peers find the peers that are physically proximate to them. To protect the privacy on Online Social Networks, a constraint can be added, in which peer A can connect to peer B only when peer A is peer B's friend or can access peer B's shared videos. The viewers of the videos S are formed into two swarms. Since the nodes in each swarm have a probability of owning chunks of the same video, they can bring chunks from their swarm-peers without having to rely on querying the server or large scale query flooding.

Step 4

Regular Tracking by server The server tracks, and keeps records 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 FaceTube's thresholds of predefined Tl and Th. The source node itself can perform this duty if it has sufficient capacity. The nodes in the system should report their video providing activities to the server. When the server confirms that a peer is a non-follower of the source node, the server it notifies the source node about the non-follower along with its interests. The server must also periodically update the roles of the followers and non-followers. Whenever a node becomes neither a follower nor a non-follower, the server removes it by notifying others to disconnect from the node. However, when a follower becomes a non-follower, its connections are also updated accordingly.

2. An Online Social Network-based chunk pre fetching algorithm:

The main aim of the algorithm is to make the videos available to the nodes without having to go back to the server.

Step 1

Push the prefix Whenever a source node uploads a brand new video to the server, it also pushes the prefix (i.e. first chunk) of the video to its follower nodes and to the interest-cluster peer nodes in the interest clusters matching the content of the video. After this, the prefix receiver nodes store the prefix in their cache. Because these followers and interest-cluster-peers might watch the video, the cached prefixes have a high probability of being used. As soon as the nodes request the videos, the locally stored prefix will be played immediately without delay.

Step 2

Retrieval of the remaining video In the meanwhile, the node will try to retrieve the remaining video chunks from its swarm-peers. A requester request 4 online nodes in parallel to provide the video content, to guarantee provider availability and achieve low delay by retrieving chunks concurrently. First, it contacts interest cluster- peers, then followers, and finally the source node. However, if the requester cannot find four providers even after the source node is contacted; it resorts to the server as its only provider. The server will be able to guarantee the availability of the video, even if the number of online users in a swarm is relatively small.

4. Simulation Results

Our simulation is based on a part of our crawled dataset, which consists of approximately 2,000 unique videos. We have selected 5,000 nodes from the trace data. The connectivity degree of the selected nodes is based on power-law distribution with a skew factor of 0.5 [6].We then assigned each one of the 2,000 videos to a randomly selected node in the system. We have used the statistics in Table 1 used in [4], to simulate real world Internet environment, where all the peers have heterogeneous bandwidth. Each node is given a location ID in [1-10], in order to simulate the geographic locations of nodes. The nodes that have numerically closer IDs are geographically closer nodes. We have used the method of distribution of friendship in spatial proximity from [7] in our experiments. Table 2 depicts the default parameters unless otherwise specified. The bit rate of the video was set to 330 kbps. Based on real world observations, we have assumed that whenever a node does not receive a chunk in time for viewing, it pauses for 3 seconds and only then resumes playback. The file size of a video was randomly selected from [20-30] MB, which is the usual size for a video with a length of 2-3 playing minutes.

Table 1: Bandwidth capacity and distribution of users

Groups	Downloading bandwidth	Uploading bandwidth	Percentage of nodes
1.	768k/s	128k/s	21.4%
2.	1536k/s	384k/s	23.3%
3.	3072k/s	768k/s	55.3%

Table 2: Experiment default parameters

Parameter	Default Value
No. of clients	5000
No. of videos	2000
No. of interest categories	19
No. of interests per client	2-4
Trace duration	40 days
Chunk size	3Mbytes
Prefix length	3 Mbytes
Server uploading bandwidth	20 Mbps
Video size	Distribution of YouTube videos
Cache size	300 Mbytes

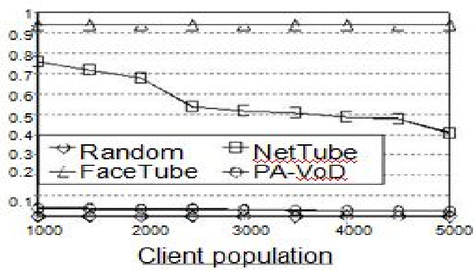

Fig. 2: Pre fetching accuracy vs. node population

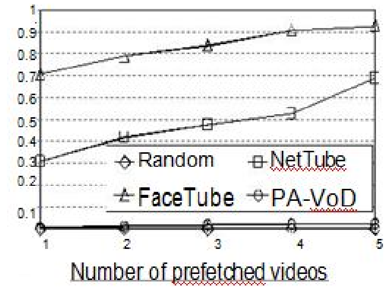

Fig. 3: Prefetching accuracy vs. no. of pre fetched videos

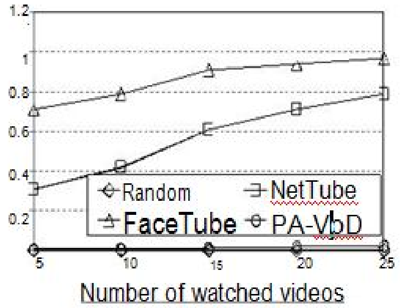

Fig. 4: Prefetching accuracy vs. no. of watched videos

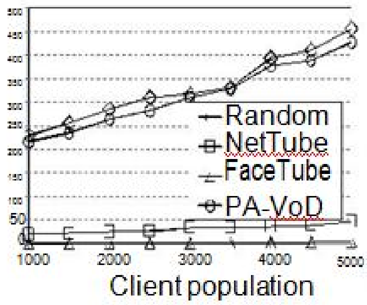

Fig.5: No. of searched clients vs. client population

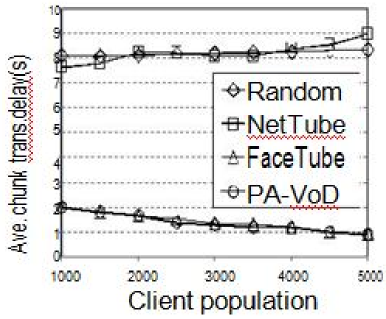

Fig. 6: Chunk transmission delay vs. client population

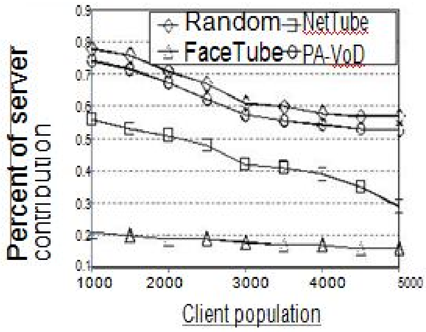

Fig. 7: Percent of server contribution vs. client population

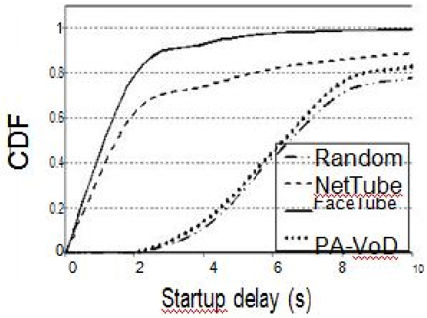

Fig. 8: CDF of node startup delay

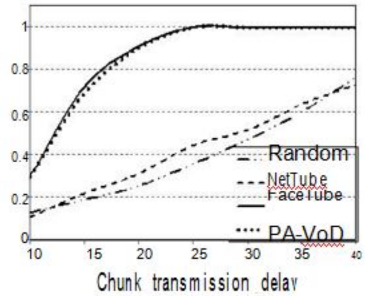

Fig. 9: CDF of chunk transmission delay

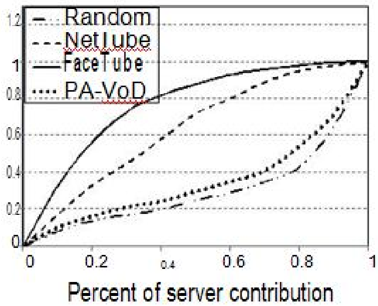

Fig. 10: CDF of the percent of server contribution in different systems

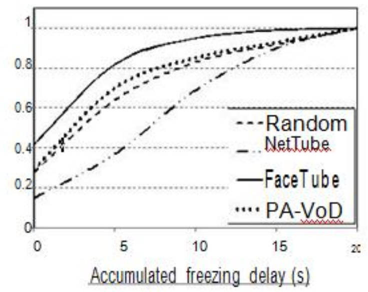

Fig. 11: CDF of accumulated freezing delay

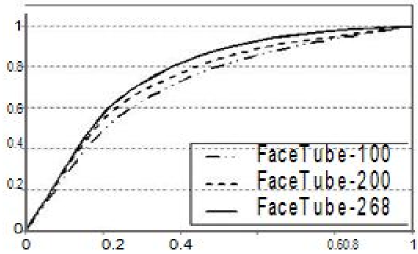

Fig. 12: CDF of the percent of server contribution in FaceTube

5. Conclusion

Video sharing is an highly popular application in OSNs. But, the client/server architecture deployed by current video sharing systems in OSNs cost a lot of resources (i.e. money, server storage, bandwidth, etc.,) for the service provider and lacks scalability. However, because of the privacy constraints in Online Social Networks, the current peer-assisted Video-on-Demand (VoD) techniques are suboptimal if not entirely applicable to the video sharing in Online Social Networks. In this paper, we have crawled video watching trace data in one of the largest online social network websites Facebook, from Mar 2013 to July 2014 and explored the users' video viewing patterns. We have found that, in a user's viewer group, 25% of the 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 relationship and interest relationship in video watching activities, we propose FaceTube, which provides efficient P2P-assisted video sharing services. Experimental and Extensive simulation results show that FaceTube can provide a low video startup delay and low server traffic demand. We have also implemented a prototype in PlanetLab to evaluate the performance of FaceTube. The experimental results from the prototype further confirm the efficiency of FaceTube.

References

[1] Facebook passes google in time spent on site for first time ever. http://www.businessinsider.com/
[2] K. Wang and C. Lin. Insight into the P2P-VoD system: Performance modeling and analysis. In Proc. of ICCCN, 2009
[3] 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
[4] X. Cheng and J. Liu. NetTube: Exploring social networks for peer-to-peer short video sharing. In Proc. of INFOCOM, 2009
[5] Haiying Shen*, Senior Member, IEEE, Ze Li, Student Member, IEEE, Yuhua Lin and Jin Li, Fellow, IEEE, "FaceTube: P2P-assisted Video Sharing in Online Social Networks", IEEE transactions on parallel and distributed systems, 2013.
[6] Nazir, S. Raza, and C. Chuah. Unveiling dacebook: a measurement study of social network based applications. In Proc. Of SIGCOMM, 2008.
[7] L. Backstrom, E. Sun, and C. Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In Proc. of WWW, pages 61–70, 2010