An Interest-based Per-Community P2P Hierarchical Structure for Short Video Sharing in the YouTube Social Network

Haiying Shen, Yuhua Lin and Harrison Chandler
Dept. of Electrical and Computer Engineering
Clemson University, SC, USA

Outline

- Introduction
- System Design
  - Trace data analysis
  - Motivation
  - Design of SocialTube
- Performance Evaluation
- Conclusions

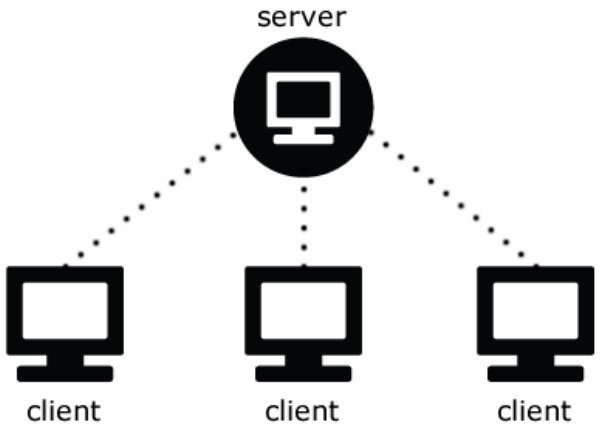
Introduction

- Video-on-demand (VoD) services are gaining popularity, e.g., YouTube, Bing Video, Vimeo, Tudou.

- Rapid increases in video content and users
- Challenge: scalability

Introduction

Client-server architecture: videos are stored and downloaded from dedicated servers



- Pros:
  - simple to implement
- Cons:
  - Prohibitive bandwidth costs for server
  - Low scalability
  - Low quality of service (QoS) during peak hours

Introduction (cont.)

P2P architecture: users download files from other users.

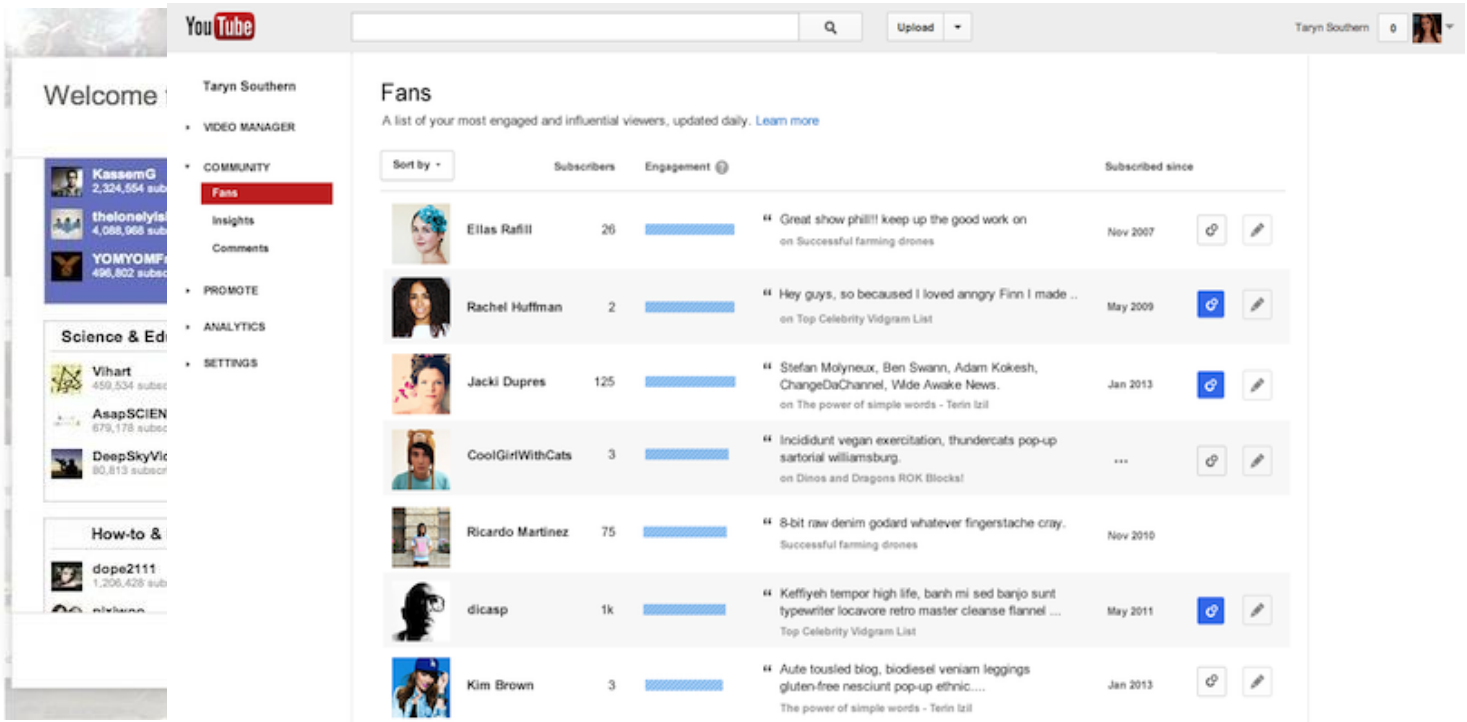
- PA-VOD [Sigcomm'07]:
  - Server directs the video request to other users currently watching the video
  - No cache: peer provider is found only when others are watching a video
- NetTube [Infocom'09]:
  - Viewers of the same video form an overlay
  - Enable users to find other videos through their neighbors
  - Cache: boost video availability

Drawback: Fail to explore social network

Introduction (cont.)

New opportunity: YouTube structure

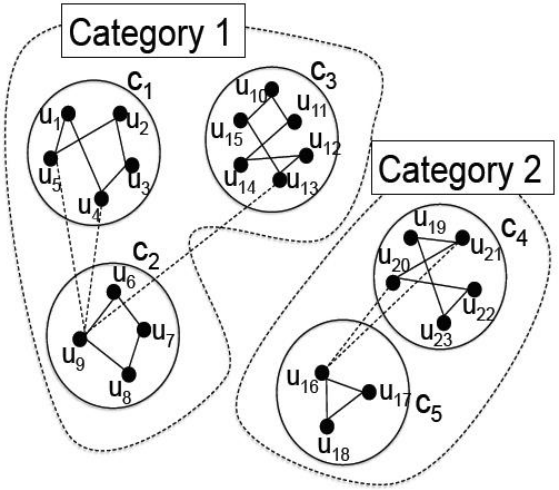
Users can subscribe to a number of categories, and each category contains a number of channels.



Reference: http://www.tubefilter.com/2013/11/04/youtube-most-susbcribed-channel/
https://support.google.com/youtube/answer/3265949?hl=en

Introduction (cont.)

Interest-based Per-Community P2P Hierarchical Structure



- Lower-level cluster: subscribers of the same channel
- Higher-level cluster: users watching channels in the same category

Introduction (cont.)

- Our contributions:
  - Analyze extensive YouTube trace data, verify the features of the YouTube social network.
  - Design Interest-based Hierarchical Structure for Short Video Sharing
  - Trace-driven simulation and PlanetLab experiments

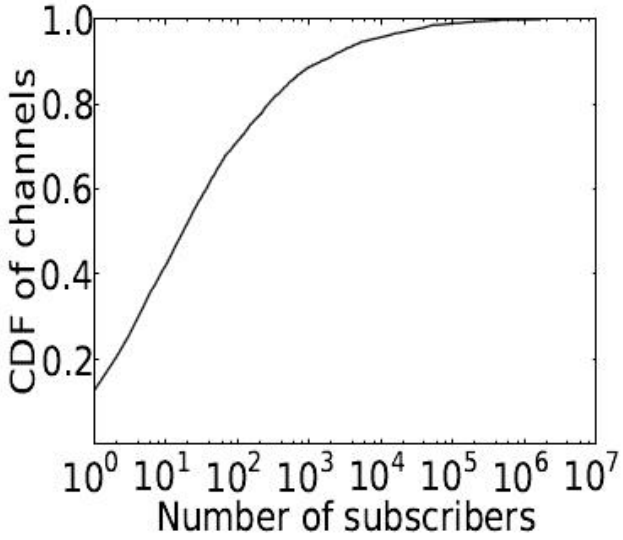
Outline

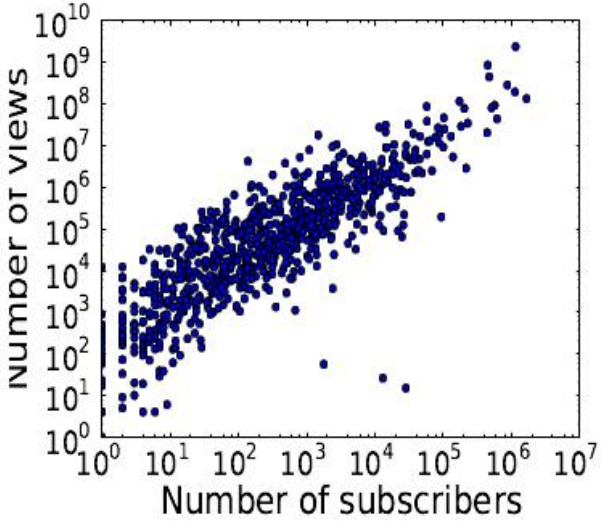
- Introduction
- System Design
  - Trace data analysis
  - Motivation
  - Design of SocialTube
- Performance Evaluation
- Conclusions

Trace data analysis

- Using the YouTube Data API
- 2,301 users,
- 261,110 videos, upload from 18 Jan. 2006 - 17 Sept. 2010

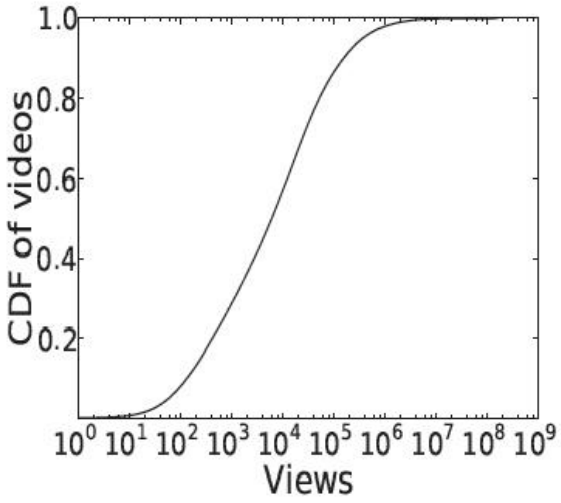
Channel popularity

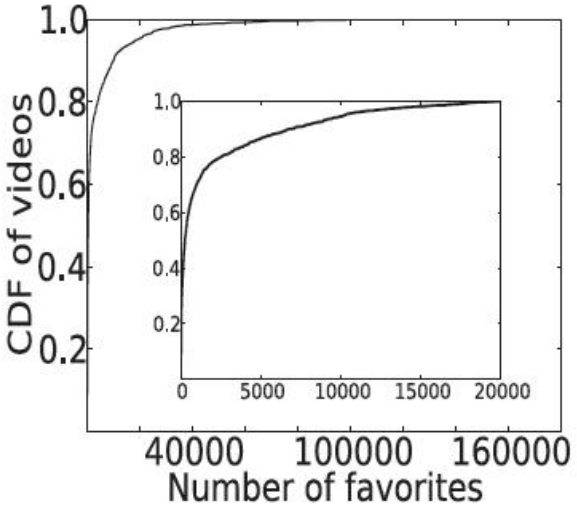




O1: Building a P2P structure based on channels would produce reduced server load and enable efficient video retrieval for users

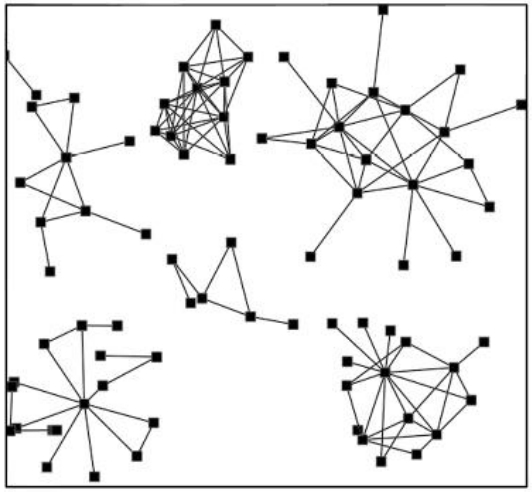
Video popularity





O2: Video view counts as a metric to determine video prefetching, thus minimizing the delay between successive video views.

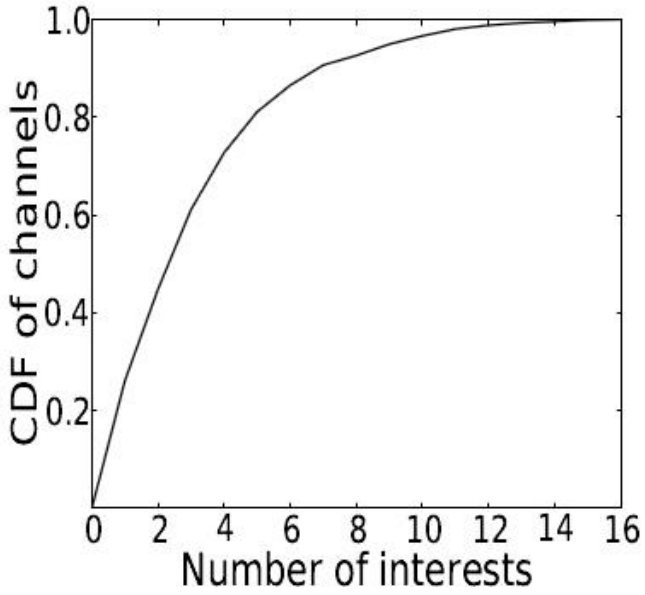
Channel Clustering and User Interest

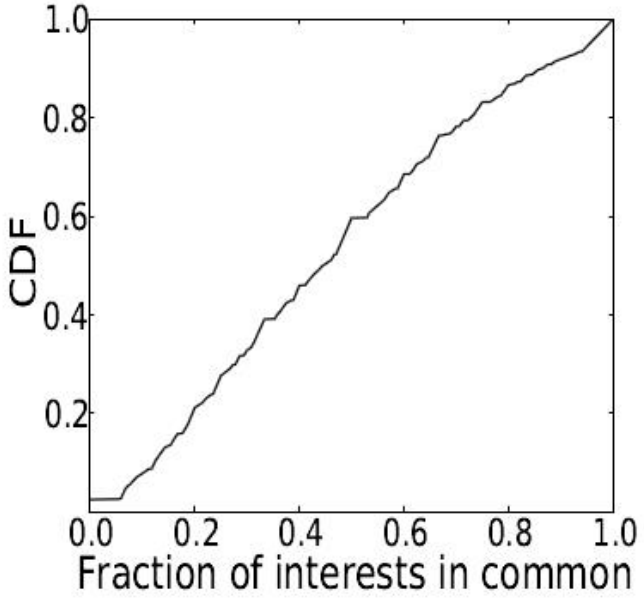


A graph of channels connected by shared users

O3: Channels have strong clustering features. Help efficiently find providers across channels in P2P short video sharing.

Channel Clustering and User Interest





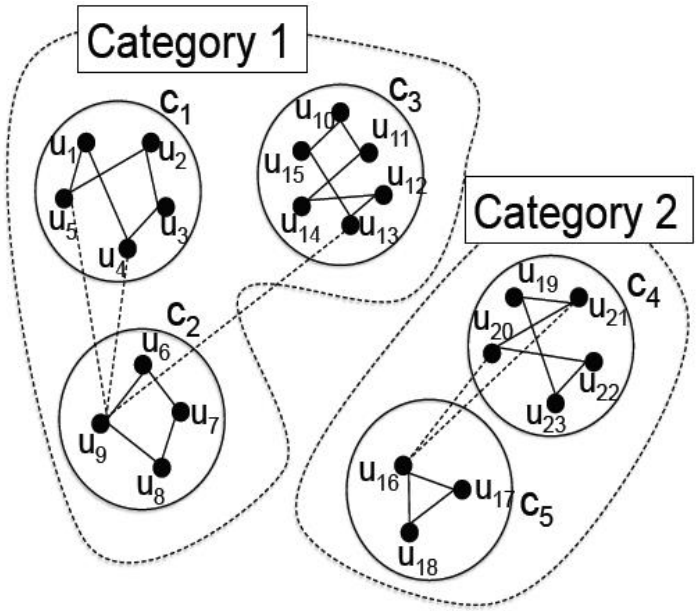
O4: Channels tend to focus on a small number of video categories; users tend to subscribe to channels that match their interests.

SocialTube

- Hierarchical per-community structure
  - lower level overlay: same-channel subscribers
  - higher level overlay: same-interest nodes (channels)
  - subscribers to the same channel and nodes sharing the same interest can share videos between each other.
- Channel-facilitated prefetching: nodes prefetch popular videos in their channels
  - minimize startup delay
  - improve video availability.

Hierarchical per-community structure

- A diagram of the network structure of SocialTube



SocialTube process

- First join: requests peers from the server
  - 1. Get a list of channel peers to C_i
  - 2. Get a list of category peers to K_i

- Peer support in video sharing:
  - 1. Sends query with TTL to C_i
  - 2. If no response in step 1, sends query with TTL to K_i
  - 3. No response in 1 and 2, request video from server

Outline

- Introduction
- System Design
  - Trace data analysis
  - Motivation
  - Design of SocialTube
- Performance Evaluation
- Conclusions

Performance Evaluation: settings

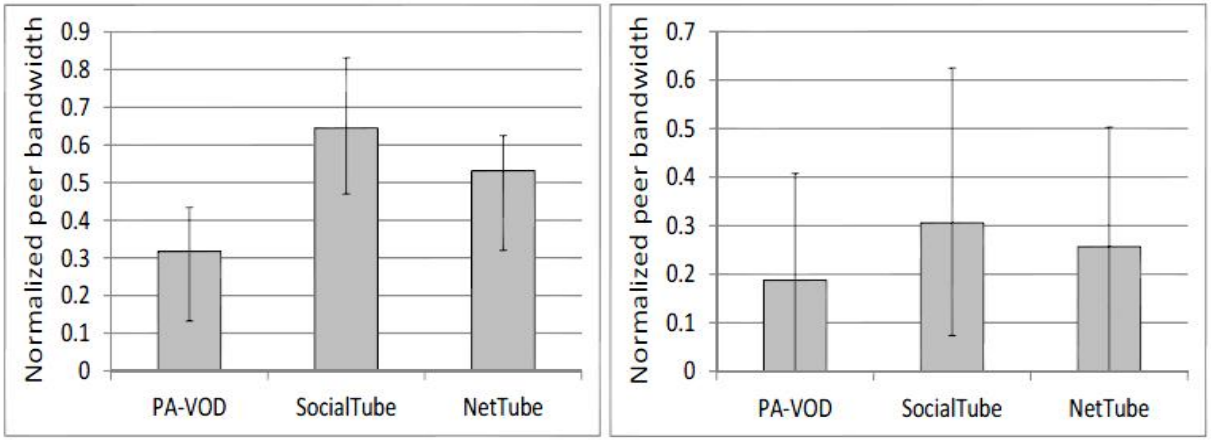
- Simulation platform: PeerSim

Parameter	Default value
Simulation duration	3 days
Number of nodes	10,000
Number of videos	10,121
Number of channels	545
Video size	YouTube video size distr.
Number of chunks per video	20
Video bitrate	320 kbps
Server bandwidth	500 mbps

- PlanetLab:
  - 250 globally distributed nodes
  - number of categories: 6
  - each category has 10 channels, and each channel has 40 videos

Performance Evaluation: results

- Server Bandwidth Reduction

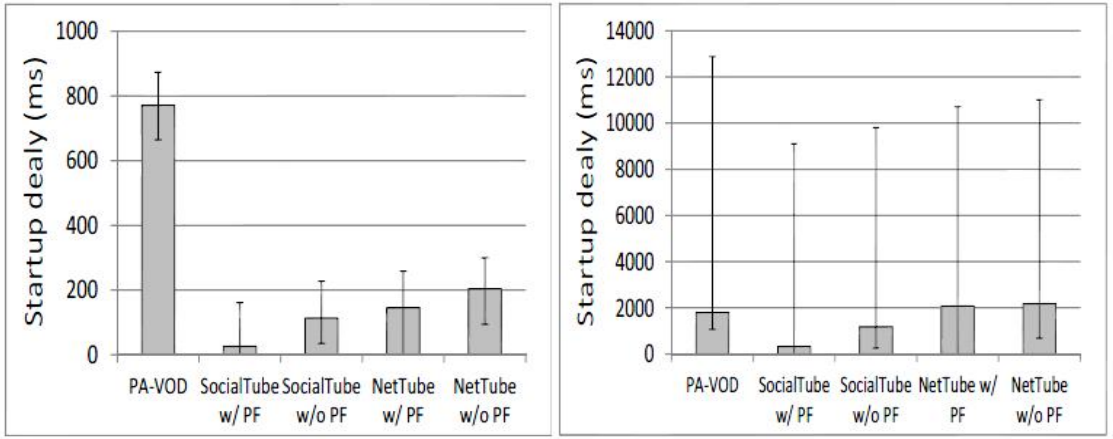


(a) The PeerSim simulator. (b) The PlanetLab real-world testbed.

- Observation: SocialTube>NetTube>PA-VoD
- Reason: in SocialTube, users have a high probability to find chunk providers in the hierarchical overlays

Performance Evaluation: results

- Startup Delay

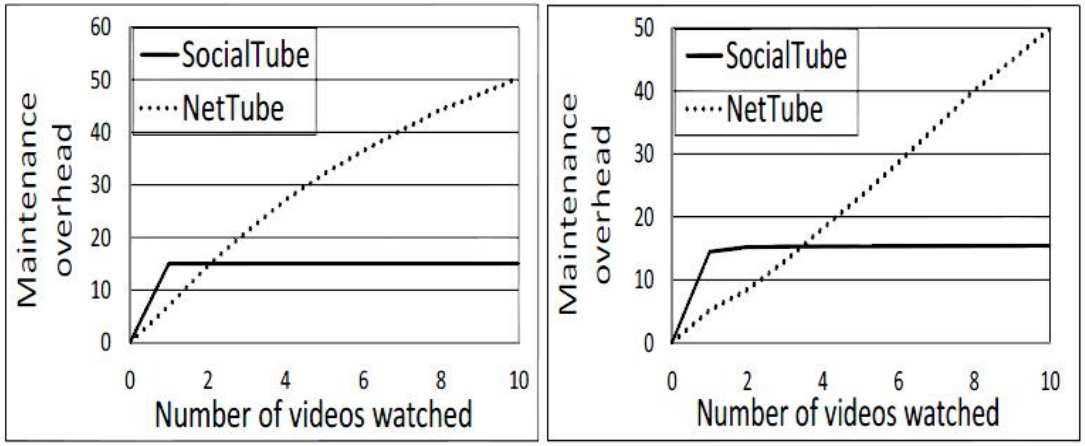


(a) The PeerSim simulator. (b) The PlanetLab real-world testbed.

- Observation: SocialTube<NetTube<PA-VoD, prefetching strategies of SocialTube and NetTube help reduce startup delay
- Reason: SocialTube reduces more requests sent to the server, avoiding overloading the server

Performance Evaluation: results

- Overlay Maintenance Overhead



(a) The PeerSim simulator. (b) The PlanetLab real-world testbed.

- Observation: SocialTube demands significantly lower maintenance overhead than NetTube
- Reason:
  - NetTube: connections a user maintains is $mlog(u)$
  - SocialTube: connections a user maintains is $log(u_c) + log(u_t)$

u: the number of users watching a video; u_c: the number of users in a channel; u_t: the total number of users within all channels in an interest

Outline

- Introduction
- System Design
  - Trace data analysis
  - Motivation
  - Design of SocialTube
- Performance Evaluation
- Conclusions

Conclusion

- SocialTube: interest-based per-community P2P system
- Trace-driven experiments on PeerSim and PlanetLab show the effectiveness of SocialTube:
  - Reduced server load
  - Low startup delay
  - Small overlay maintenance overhead
- Future work: tradeoff between maintenance overhead and availability of peer video providers

Thank you!

Questions & Comments?

Haiying Shen

shenh@clemson.edu

Associate Professor of Electrical and Computer Engineering

Clemson University