 ISSN: 2321-8169
4409 - 4414

An Effective Peer to Peer Video Sharing Scheme with Social Reciprocity

Mrs. Hemlata Sawant
Department of Computer Engineering
Alard College of engineering
Savitribai phule Pune University
Marunje, Pune 58, INDIA
hema10dec@gmail.com

Prof. Sonali Patil
Department of Computer Engineering
Alard College of engineering
Savitribai phule Pune University
Marunje, Pune 58, INDIA
sonalipatil@live.com

***Abstract*—** online video sharing and social networking are self-fertilizing speedily in today's Internet. Online social network users are flooding more video contents among each other. A fascinating development as it is, the operational challenge in previous video streaming systems persists, i.e., the large server load required for topping of the systems. Exploring the unique advantages of a social networking based video streaming system; it advocate utilizing social reciprocities among peers with social relationships for efficient involvement incentivization and development, so as to enable high quality video streaming with low server cost. Then why only video: because more people prefer watching videos. Videos induce people to stay longer on websites. People remember videos. It achievement social reciprocity with two give-and-take ratios at each peer: (1) peer contribution ratio (PCR), which calculates the reciprocity level between a couple of social friends, and (2) system contribution ratio (SCR), which records the give-and-take level of the user to & from the entire system. It expect efficient Peer to Peer mechanisms for video streaming using the two ratios, where each user optimally chooses which other users to seek relay help from and help in relaying video streams, respectively, based on combined evaluations of their social relationship and historical reciprocity levels. This design helps to gain effective incentives for resource contribution, load balancing among relay peers, and efficient social-aware resource scheduling, security to the videos and high prefetching accuracy.

Keywords- *video streaming; social reciprocity; peer to peer network*

*****

I. INTRODUCTION

Current years have seen the come into bud of online social networks (e.g., Facebook [1], Google+ [2]) and video streaming sites (e.g., YouTube [3]), as well as a coming together between the two types of systems. More and more media contents (video clips, images, etc.) are published and shared among users on social network sites while the video streaming systems are increasingly leveraging social networks to promote their videos and attract viewers. Peer-to-peer (P2P) technology has been encouraged to improve the server load in video streaming applications, such that users (peers) directly send video streams to each other, with less dependence on the dedicated servers. Challenges remain in a peer to peer design, among which incentivizing sufficient and stable peer bandwidth contribution has been a fundamental one. There have been a number of P2P incentive designs based on direct or indirect resource trading, but none of them has utilized social connections among the peers. The unique setting of a social video streaming system has made very promising a more effective social reciprocity based incentive for P2P video streaming over a social network, that exploits the natural intentionality for each peer to help socially connected peers. In this paper, we design a social video streaming system which utilizes social reciprocity to incentivize effective bandwidth contribution and scheduling at the users, and employ peer to peer design to distribute video streams with low server cost. To peer to peer video streaming, we consider resource Streaming between friends and between non-friends. A user (peer) in our system is responsible to send its own video stream to all other interested users, either directly if it has Sufficient upload bandwidth, or by resorting to relay helpers when there are too many viewers.

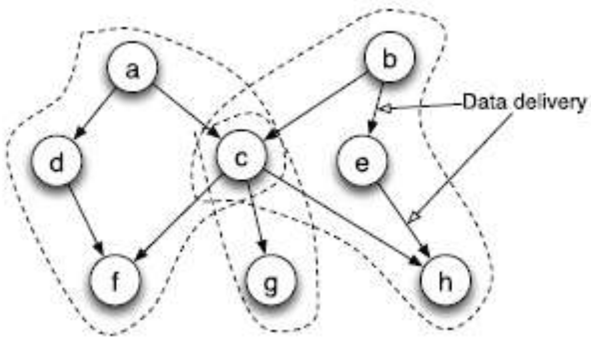

Figure 1: An illustration of peer to peer video Streaming: multiple videos.

In the example scenario in figure 1, a, b and c are 3 source peers streaming 3 videos, which we assume have the same streaming rate, respectively. We have viewers d, f ∈ Va, g ∈ Vc, and e, h ∈ Vb, relay helpers c, d ∈ Ra, c, e ∈ Rb, and Rc ={a, b}. Here peer c plays the roles of source peer and relay peer simultaneously.

Our key design is to effectively incentivize peers with extra upload bandwidth to serve as relay helpers in others stream distribution, for which we exploit social reciprocities. We define two light-weighted give-and-take ratios at each peer: (1) peer contribution ratio (PCR), which evaluates the historical reciprocity level between pair of social friends, and (2) system contribution ratio (SCR), that records the give-and-take level of the user to and from the entire network. A social reciprocity index (RI) is defined at each user to evaluate each other peer, based on the two give-and-take ratios and the strength of relationship between them two. This index is used in the design of two efficient algorithms for each user to optimally decide which other users to seek relay help from and help in relaying video streams, respectively.

Next we are providing to security to the video stream for that purpose we use advanced encryption standard (AES) algorithm.

In social networking site efficiency is most important thing. So improve efficiency we use prefetching strategy. A prefetching strategy to enable high worth streaming. As an extension to this work we are using public and private video streaming to maintain the privacy of videos.

Our design is able to achieve the following effectiveness: (1) peers are maximally incentivized to contribute their available upload resources, (2) load on different relay peers is effectively balanced, and (3) upload bandwidth in the system is efficiently scheduled with social awareness (or social preference), in that users are more inclined to seek help from and help other users with closer social ties (4) Security to the video stream (5) high accuracy.

The rest of this paper is organized as follows. We discuss related literature in Sec. 2, and detailed algorithm design in Sec. 3, and present our proposed model in sec 4. Next Result in sec. 5. Finally, we conclude the paper in Sec. 6.

II. RELATED WORK

In 2013, Shen [5] implemented SocialTube, a novel peer-assisted video sharing system that explores social relationship, interest similarity, and physical location between peers in (online social network) OSNs. Specifically, SocialTube incorporates four algorithms: a social network (SN) based P2P overlay construction algorithm, a SN-based chunk prefetching algorithm, chunk delivery and scheduling algorithm, and a buffer management algorithm.

In 2012, in Wang's study [6] they verify that large-scale measurement of a real world online social network system to study the propagation of the social video contents. They shortened important characteristics from the video propagation patterns, including social locality, geographical locality and temporal locality. Motivated by the measurement insights, they intend a propagation based social aware replication framework using a hybrid edge cloud and peer assisted architecture, namely PSAR, to serve the social video contents. This replication strategies in PSAR are based on the design of three propagation based replication indices, including a geographic influence index and a content propagation index to guide how the edge-cloud servers backup the videos, and a social influence index to guide how peers cache the videos for their friends. By incorporating these replication indices into system design, PSAR has signicantly improved the replication performance and the video service quality.

In 2010, Recently Liu [8] carry out private (peer to peer) P2P file sharing systems with three contributions. First measurement study on a representative private Bit Torrent site provides more incentive for users to contribute and seed. Second, Liu developed a game theoretic model and analytically show that the ratio mechanism indeed provides effective incentives. But existing ratio incentives in private sites are at risk to collusions. Third, to prevent collusion, proposed an upload entropy scheme, and show through analysis and experiment that the entropy scheme successfully limits colluding, while rarely affecting normal users who do not collude.

In 2010, Liu [10] proposed a new incentive paradigm, Networked Asynchronous Bilateral Trading (NABT), which can be applied to a broad range of P2P applications. In NABT, peers belong to an underlying social network, and each pair of friends keeps track of a credit balance between them. There are only credit balances maintained between pairs of friends. NABT allows peers to supply each other asynchronously and further allows peers to trade with remote peers through intermediaries.

In 2009 Cheng [12] and Liu addressed NetTube, a peer to peer delivering framework that discovers the clustering in social networks for short video sharing, including a series of key design issues to realize the system, and a bi-layer overlay, an efficient indexing scheme and a prefetching strategy leveraging social networks.

In 2006, Pouwelse and Wang [16] implemented TRIBLER P2P file-sharing system, it's an as a set of extensions to Bit torrent. TRIBLER helps to automatically build a robust semantic and social overlay on top of Bit torrent, one of the most popular P2P file-sharing systems. They have shown how various TRIBLER components can yield good performance with respect to existing solutions. In particular, Pouwelse and Wang have presented evidence that collaborative downloading yields a significant speedup when used in a real Bit torrent swarm

In 2005 X. Zhang [18] presented Coolstreaming DONet, a Data-driven Overlay Network for live media streaming. The fundamental operations in DONet are very simple like every node periodically exchanges data availability information with a set of partners, and retrieves unavailable data from one or more partners, or supplies available data to partners.

In 2003, Kien A. Hua [19] designed a peer-to-peer technique called ZIGZAG for single-source media streaming. ZIGZAG allows the media server to distribute content to many clients by organizing them into an appropriate tree rooted at the server. This application layer multicast tree has a height logarithmic with the number of clients and a node degree bounded by a constant. This helps reduce the number of processing hops on the delivery path to a client while avoiding network bottleneck. Consequently, the end-to-end delay is kept small. Although one could build a tree satisfying such properties easily, an efficient control protocol between the nodes must be in place to maintain the tree under the effects of network dynamics and unpredictable client behaviors. ZIGZAG handles such situations gracefully requiring a constant amortized control overhead. Especially, failure recovery can be done regionally with little impact on the existing clients and mostly no burden on the server.

III. DETAILED DESIGN

A. *Problem Definition*

Now a day's online video streaming and social networking are rapidly crossing each other. So content sharing on social networking users is increased, but the operational challenges are still continues like the huge amount of server cost required climbing the system. A huge amount of server bandwidth is expected, in order to distribute the large volumes of videos generated and uploaded by users. Minimum coordination among friends in social networks as well as less accuracy and high startup delay.

 ISSN: 2321-8169
4409 - 4414

B. *Social Reciprocity*

A social reciprocity index for each peer to evaluate its perceived contribution level from another peer, based on the two give-and-take ratios and the strength of social relationship between them two, as follows:

1) *Social Reciprocity Index (RI):* A social reciprocity index $e_A(B)$ for each peer A to evaluate its perceived contribution level from another peer B, based on the two give-and-take ratios and the strength of social relationship between them two, as follows:

$$e_A (B) = (1-f_{AB}) W_B + f_{AB} W_A(B) \qquad (1)$$

2) *Peer contribution ratio (PCR):* Peer contribution ratio evaluates the give-and-take balance between two social friends. PCR $W_B (A)$ is defined as the ratio of peer A's upload contribution to peer B over the total mutual contributions between the two:

$$W_B (A) = C_B (A) \;/\; C_A(B) + C_B (A) \qquad (2)$$

3) *System contribution ratio (SCR):* To evaluate each peer's contribution in the entire system, we also define a system contribution ratio W_A as follows:

$$W_A = y_A' / \; y_A' + y_A \qquad (3)$$

This index is used in two effective algorithm for each source to choose which other peers to seek relay help from, and for each relay to optimally decide which source peers to help, respectively. Specifically, each source will choose peers with smaller RI values it evaluates as relay helpers, and each relay be likely to help sources with larger RI values it has evaluated. Table I summarizes notations of the system.

TABLE 1: NOTATIONS

Symbol	*Definition*
PCR	Peer contribution ratio
SCR	System contribution ratio
RI	Social Reciprocity index
y_A'	The amount of upload resource peer A has contributed to the system.
y_A	The amount of upload resource others have provided to A.
$C_A(B)$	The amount of upload resource peer B has provided to A
W_A	The system contribution ratio of peer A.
$W_A(B)$	The peer contribution ratio of peer B between the pair of peer A and B.
f_{AB}	The social closeness between peer B and peer A
$e_A(B)$	The social reciprocity index of peer B evaluated by peer A
V_A	The viewer set of source peer A
R_A	The relay set of source peer A
S_B	The requesting source set of relay peer B
u_A	The upload capacity of relay peer A
r_A	The stream rate of the video generated by A
$X_{AB}^{(T)}$	The number of viewers that A asks relay B to serve in time slot T
$a_{AB}^{(T)}$	The number of viewers that relay B is to serve for source peer A in time slot T
L	The number of candidate relay peers provided by the tracker server.
K	The maximum number of relay peers a source maintains
T_h	The duration of a time slot

C. *Source Algorithm*

The function of source algorithm is to select relay. When the number of viewers of the video source peer produces exceeds its upload capacity, peer search for relay helpers. Three steps are involved: (1) Selecting Relays: When a source peer asks the server for candidate relays, the server will assign it with the relay peers that are socially closer to its friends. (2) Estimating Relay's Available Upload Bandwidth: Each candidate relay peer, which has unused bandwidth, may possibly help multiple other source peers. (3) Assigning Viewers to Relays: After relay peers are selected, the source peer decides which viewers to be served. The main algorithm working is as follows:

Input: Video stream
Output: Post video stream viewer or request to relay
Steps:

1. Start procedure
2. IF $u_A \geqslant V_A \mid r_A$ then
3. A will distribute the video stream to all viewers directly
4. Else sort RA in ascending order of RI's (e_A(B)'s).
5. Let y denote the number of viewers to be served by relay peers.
6. For all relay peer B in sorted list R_A do
7. If A is able to distribute to the y viewers or its upload bandwidth is smaller than 2rA then break
8. End if
9. Estimate the relays available upload bandwidth
10. Then min viewers are assigned to relay peer B
11. Send to relay B the video stream together with the list of assigned viewers.
12. End for
13. Send the video stream to remaining viewers one by one until A's remaining upload bandwidth is smaller than 2rA
14. Resort the server for relaying to all remaining viewers.
15. End if
16. End procedure

D. *Relay Algorithm*

This relay algorithm is used for resource contribution. First a peer has extra service capacity it may roll itself as a relay with the server, and may take itself down from the candidate list when its upload bandwidth is fully used. The steps of algorithm is as follows:

Input: Request from source or viewer
Output: Post video stream to viewer
Steps:

1. Start procedure
2. $u \leftarrow u_B - \mid V_B \mid r_B$
3. Sort source in SB in descending order of RI (e_B (A)'s)
4. For all source peer A is sorted list S_B do
5. Then find out number of viewers that relay B is to serve for source peer A in time slot T.
6. End for
7. Send upload allocation to source peer A according to S_B
8. On receiving video requests from A's viewers, B serves videos to corresponding viewers.
9. End procedure

 ISSN: 2321-8169
4409 - 4414

IV. Proposed Model

A. *Mathematical model*

Relevant Mathematics Associated with the Dissertation:
Set Theory:
1) Identify users
I = {i1, i2, i3, i4….}
Where I is main set of users
2) Identify their Registration
R = {r1, r2, r3, r4….}
Where R is main set of Registration
3) Identify Servers
S = {s1, s2, s3, s4….}
Where S is main set of servers
4) Process:
We define a security property. To ensure the correctness of user uses the application we propose login-password protection. Identify the processes as P.
P= {Set of processes}
P = {P1, P2, P3, P4……Pn}
P1 = {e1, e2}
{e1 register user}
{e2 identify user during login}
P2= {e1, e2, e3, e4}
{e1= Incorrect Password}
{e2= Correct Password}

B. *System Architecture*

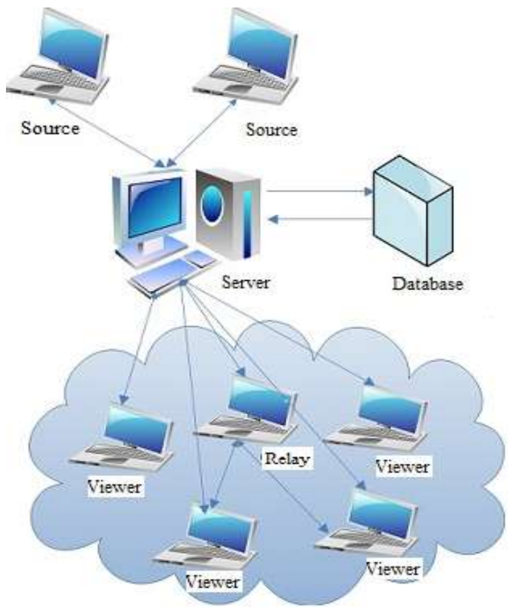

Figure 2: Proposed System

This scenario involves the Source algorithm and relay and viewer for effective resource allocation. There are three main phases for the system source, relay and viewer.
 Source algorithm:
The function of source algorithm is to select relay. When the number of viewers of the video source peer produces exceeds its upload capacity, peer search for relay helpers.
 Relay algorithm:
This relay algorithm is used for resource contribution. First a peer has extra service capacity it may roll itself as a relay with the server, and may take itself down from the candidate list when its upload bandwidth is fully used.
 Viewer:
User wants to see or play video through his own system then he will act as viewer. Viewer can send request to source as well for the particular video.

The security is the main issue in this solving the problem. The security is provided by AES algorithm and prefetching approach to reduce interruption. Hence to progress in this area it is designed to develop prefetching strategy and AES algorithm to provide security to the network. The block diagram of proposed system is shown in Figure 2.

C. *Prefetching strategy*

This prefetching strategy is help to improve efficiency. A prefetching strategy to enable high worth streaming. By prefetching the prefix chunks of a video that is likely to be viewed by a user, a short startup delay and smooth playback can be achieved. This Prefetching strategy is based on users preferences gradually learnt from their historical video access patterns. In the online social network, user's preferences of videos are predicted by:

Choosing Videos to Prefetch: In the social video streaming, a viewing user has to choose among many videos to Prefetch. In our design, the viewing user's video preference is predicted by, Historical Selection of the Source User. For example In the online social network, viewing user A chooses videos shared by multiple source users in set Y_A, which is the set of source users whose shared videos are likely to be viewed by user A. Y_A contains all A's friends, and strangers whose shared videos have been watched by A before. We define a pairwise index hAB to evaluate A's historical selection level of source user B as

$$h_{Ak} = C_{AB} / \Sigma\ K \in Y_A C_{Ak} \quad (4)$$

Where $B \in Y_A$ and C_{AB} is the number of historical views that peer A watches videos shared by peer B. Large h_{AB} indicates that user A prefers videos shared by user B, such that user A can still choose videos shared by user B in the future.

Social Closeness between the Viewing User and the Source User: Users are more likely to watch videos shared by their friends. Such social preference between friends is evaluated by a straightforward metric, which is the fraction of their common friends over their total friends. The social closeness f_{AB} between user A and user B is defined as

$$F_{AB} = | f_A \cap f_B | \ / \ | f_A \cup f_B | \quad (5)$$

Where intuitively, larger fAB indicates stronger social closeness between A and B.

Preference of a Source User: According to our observations, both historical selection and social closeness predict a viewing user's future preference of source users. A source preference index eAB to evaluate how much viewing user A likes to watch videos shared by source user B as

$$e_{AB} = w_A f_{AB} + (1- w_A)\, h_{AB} \quad (6)$$

Where w_A is a parameter for a peer to adjust the weights of historical selection and social closeness. Larger w_A indicates that A selects videos more according to friend links.

Video Popularity: Source preference index e_{AB} evaluates viewer A's preference to choose source user B. However, a source can share multiple videos; while for a viewing user, only being able to select the best source users is not enough to do the prefetching. We next give the video rank scheme based on video popularities and source preference indices. The popularity of a video reflects how much viewing users like the video. The popularity of video k is defined as p_k, which is the number of views that users watch the video. When a user logs in his account, he is provided a list of unwatched videos shared by his friends and other users. The number of these videos can be very large for a viewing user.

After that, the user (peer) downloads prefix chunks of the ranked videos in order. When prefetching a video, a peer actively downloads the prefix chunks of that video, i.e., the first several chunks prefetching strategy is carried out by a peer locally when the buffered chunks for the current video can be played. The startup delays thus can be reduced when the user plays videos that have been prefetched.

D. Advanced Encryption Standard (AES)

Why Advanced Encryption Standard (AES)?

TABLE 2: COMPARISON OF ENCRYPTION ALGORITHMS

	DES	*AES*	*RSA*
Complexity	Complex	Complex	Simple
Speed	High	High	High
Memory	N/A	Very Low	N/A
Key Type	Private	Private	Public
Key Length	56 Bit	128, 192,256 bit	Variable
Security Level	Low	High	High

AES comprises three block ciphers, AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128-, 192- and 256-bits, respectively. Symmetric or secret-key ciphers use the same key for encrypting and decrypting, so both the sender and the receiver must know and use the same secret key. All key lengths are deemed sufficient to protect classified information up to the Secret level with Top Secret information requiring either 192- or 256-bit key lengths.

The constructed Advanced encryption standard (AES) encryption is generally used for encrypt the videos. AES is a feasible solution to secure real time video or video on demand transmissions.

E. Video Privacy

With the rise of social networks and with more and more organizations offering their videos, photos online, consumers are becoming progressively more comfortable divulging personal information to perform transactions. Unfortunately, fraudsters have discovered how to exploit this for their own benefit. As a result, businesses must take additional steps to protect privacy and educate their customers on what they are doing to prevent fraud. Privacy of video stream is another important aspect in social networking application. For example in confidential data. Like. There are three friends Ajay, Vijay and Sujay. Generally these three friends share videos with each other. In between these two friends Ajay and Vijay are very close friends. And Ajay wants to share his personal video clip with Vijay. But Ajay also wants to that Vijay will never share video. That means Ajay wants to maintain privacy with his video clip and Vijay should watch video because he is very close friend of Ajay. For that purpose we have designed a constraint where Ajay will choose the private option button for video sharing. And in another time Vijay only can watch video. Vijay cannot forward or share with other.

As a process we developed two options for public and private. Then user will choose whatever he wants. In this we will get user satisfaction. It also helps to attract users because of privacy constraint. It also helps to increase the relationship between two friends (users).

V. RESULT

A. Experimental Settings

We have implemented a prototype peer to peer network video streaming system in JAVA programming language, and deployed it on Apache tomcat server. 200 users are used in our experiments, corresponding to the source peers, relay peers, and viewers. Peers in the system share and view videos. We follow the distribution of videos that are uploaded by users into database, to emulate new sharing of video by source peers in each time slot in our system. We follow the distribution of videos that are viewed by users on to emulate how viewer peers select the videos to watch in our system. We also follow the social graph to calculate the social closeness of friends in our experiments.

B. Graph

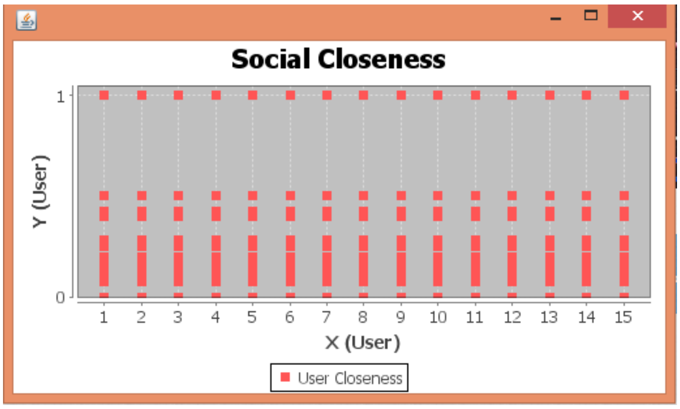

Figure 3 shows the social closeness graph gives closeness between friends

If social closeness is 0 then no existing relationship and social closeness is 1 then strong relationship. That means social closeness is more then it helps to improve time efficiency and to reduce startup delay.

Figure 4 shows the time graph which calculates the time between sender and receiver sharing of friends

VI. Conclusion

Social video streaming Network security is the today's biggest challenge. This paper presents social reciprocity with two light weighted give-and-take ratios at each peer with message sharing. Deeds the natural intentionality for each peer to help socially connected peers and strangers also. Here every time source peer is responsible to send video stream. For that purpose source and relay uses social reciprocity index at every transaction. Then each source will choose peers with smaller RI values it evaluates as relay helpers, and each relay tends to help sources with larger RI values it has evaluated. The streaming services are asymmetric between the sender and the receiver, since the video stream only flows in one direction, from a server to destination.

As an extension, it is also desirable to develop additional incentive strategies that maximally motivate intermediaries to participate in system. Effectively isolate cheaters and motivate peers to cooperate in service by providing AES encryption and decryption and also keeping small delay to smooth playback.

Acknowledgment

I would like to express my sincere thanks of gratitude to my esteemed guide Prof. Sonali Patil for her guidance and suggestions during this work. I am really thankful to her. I am thankful to all who have directly or indirectly guided and helped me in delivery of this work. Finally I would also like to thank friends who helped me a lot in finishing this work within the limited time.

References

[1] Social networking site, http://www.facebook.com.

[2] Social media sharing, http://plus.google.com.

[3] Social Video sharing, http://www.youtube.com.

[4] Z. Wang, C. Wu, L. Sun, and S. Yang, "Peer-Assisted Social media streaming With Social Reciprocity," in Proc. of ACM/IEEE, 2013.

[5] Haiying Shen, Ze Li, Yuhua Lin and Jin Li, "SocialTube: P2P-assisted Video Sharing in Online Social Networks", VOL: PP NO: 99 YEAR 2013.

[6] Z. Wang, L. Sun, X. Chen, W. Zhu, J. Liu, M. Chen, and S. Yang, "Propagation-Based Social-Aware Replication for Social Video Contents," in Proc. of ACM Multimedia, 2012.

[7] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media? ", Proc 2010 ACM WWW.

[8] Z. Liu, P. Dhungel, D. Wu, C. Zhang, and K. Ross, "Understanding and improving ratio incentives in private communities", in Proc. 2010 IEEE ICDCS.

[9] K. Lai and D. Wang, "Towards Understanding the External Links of Video Sharing Sites: Measurement and Analysis", in Proc. of ACM NOSSDAV, 2010.

[10] Z. Liu, H. Hu, Y. Liu, K. Ross, Y. Wang, and M. Mobius, "P2P Trading in Social Networks: The Value of Staying Connected," in Proc. of IEEE INFOCOM, 2010.

[11] Q. Li, S. Zhu, and G. Cao, "Routing in Socially Selfish Delay Tolerant Networks," in Proc. of IEEE INFOCOM, 2010.

[12] X. Cheng and J. Liu. "Nettube: Exploring social networks for peer to peer short video sharing", in Proc. of IEEE INFOCOM, 2009.

[13] Huiqi Zhang, Ram Dantu, Joao Cangussu .,"Quantifying Reciprocity in Social Networks" ,2009 International Conference on Computational Science and Engineering.

[14] Mei-Hsuan Lu, "Optimizing Transmission for Wireless Video Streaming," Ph.D Thesis, Department of Electrical and Computer Engineering of Carnegie Mellon University, July 2009.

[15] Y.Hu Huang, T. Fu, D. Chiu, J. Lui, and C. Huang, "Challenges, Design and Analysis of A Large-scale P2P-VoD System," in Proc. of ACM SIGCOMM, 2008

[16] J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D.Epema, M. Reinders, M. Van Steen, and H. Sips, "Tribler: a social based peer-to-peer system," Concurrency and Computation: Practice and Experience, vol. 20, no. 2, pp. 127–138, 2008.

[17] Lian Jian_ and Jeffrey MacKie-Mason, "Why Share in Peer-to-Peer Networks?" May 26, 2006.

[18] Xinyan Zhang, Jiangchuan Liuy, Bo Liz, and Tak-Shing Peter Yum, "CoolStreaming/DONet: A Data-Driven Overlay Network for Efficient Live Media Streaming", May 30 2004.

[19] Duc A. Tran, Kien A. Hua, Tai Do, "ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming", IEEE INFOCOM 2003.

[20] Philippe Golle Kevin Leyton-Brown Ilya Mironov, "Incentives for Sharing in Peer-to-Peer Networks", Computer Science Department Stanford University.

[21] Michal Feldman and John Chuang, "Overcoming Free-Riding Behavior in Peer-to-Peer Systems", School of Information Management and Systems University of California, Berkele.

[22] Zhi Wang, Lifeng Sun, Chuan Wu, and Shiqiang Yang. "Guiding Internet-Scale Video Service Deployment Using Microblog-Based Prediction".