SOCIAL MEDIA CONTENT DISTRIBUTION: MEASUREMENT AND ENHANCEMENT

by

Xu Cheng

B.Sc., Peking University, 2006

M.Sc., Simon Fraser University, 2008

A Thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2012

APPROVAL

Name:	Xu Cheng
Degree:	Doctor of Philosophy
Title of Thesis:	Social Media Content Distribution: Measurement and Enhancement
Examining Committee:	Dr. Pavol Hell Chair
	Dr. Jiangchuan Liu, Senior Supervisor Associate Professor
	Dr. Jian Pei, Supervisor Professor
	Dr. Ivan V. Bajić, SFU Examiner Associate Professor, Engineering Science
	Dr. Jianping Pan, External Examiner Associate Professor, Computer Science, University of Victoria
Date Approved:	June 8, 2012

Partial Copyright Licence

Abstract

In the past decade, such popular social media as YouTube, Facebook, and Twitter have substantially changed the content distribution landscape and become an important part in people's everyday life. Extensive research works have been conducted to understand them in the recent years. However, a number of new features emerge and a number of directions are yet to be explored.

This thesis largely extends the current research efforts on social media content distribution by measurements and enhancements. We first analyze YouTube Insight dataset from a partner's view, revealing the inherent relationship among various metrics which affect the popularity of the videos. Our findings facilitate YouTube partners to adapt their content deployment and user engagement strategies to generate more views and subsequently increasing their revenues. We also take an important step towards understanding the characteristics of video spreading in social media, examining the user behaviour and the spreading structure. We propose an epidemic model to capture the process of video spreading, which serves as a valuable tool for workload synthesis, traffic prediction, and resource provisioning.

Motivated by our measurement and a user questionnaire survey, we reveal a new scenario of coexistence of sharing and streaming. We propose a novel system that leverages stable storage users and yet inherently prioritizes living streaming flows, providing better scalability, robustness, and streaming quality. On the other hand, the recently emerged cloud service is a promising solution to the huge demands of bandwidth and storage from the growing social media. However the existing works on partitioning social media contents only focus on preserving the social relationship. We take an important factor, user access pattern, into account, and formulate the problem as a constrained k-medoids clustering problem. Our solution shows significant decrease of the access deviation and flexible preservation of the social relationship.

To my family
R.I.P., grandma

"No Pain, No Gain"

Acknowledgments

First, I give my foremost gratitude to my senior supervisor Dr. Jiangchuan Liu. His significant enlightenment, guidance and encouragement on my research are invaluable for the success of my PhD as well as Master's Degree in the past six years. His insights and suggestions have made the road to completing this thesis smoother. I would not have finished this thesis without him.

I thank my supervisor Dr. Jian Pei, for his instructions and comments that helped me complete this thesis. His visionary thoughts and serious working style have influenced me greatly.

My gratitude also goes to Dr. Ivan V. Bajić and Dr. Jianping Pan for serving on the thesis examining committee. I thank them for their precious time reviewing my thesis and for advising me on improving this thesis. I would like to thank Dr. Pavol Hell for chairing my PhD thesis defence. I would also like to extend my gratitude to the faculties and staffs in the School of Computing Science at Simon Fraser University.

I thank Ms. Shahrzad Rafati and Dr. Mehrdad Fatourechi, for providing me the opportunity of the internship at BroadbandTV Corp. The collaboration was enjoyable and contributed greatly to this thesis. I am looking forward to starting my career at BroadbandTV.

I thank my colleagues and friends at Simon Fraser University, as well as ex-colleagues whom I worked with. Although I am not listing your names here, I am deeply indebted to you.

Last but certainly not least, I thank my family for their love, care, and support: my parents, my grandparents, my aunts and uncles, my brother Lei, and my dearest wife Jia. I sincerely hope that I have made them proud of my achievements today. This thesis is dedicated to you all. I love you all!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the past decade, a number of web technologies such as Ajax (Asynchronous JavaScript and XML) [48], Adobe Flex [1], JSON (JavaScript Object Notation) [32], and others, have significantly contributed to the development of Web 2.0, which was formally introduced in the late 2004 [79]. The advanced interaction with webpages provided by the Web 2.0 technologies allows users to generate and provide contents. This behaviour has substantially changed the conventional Web 1.0, where users just retrieve information. Nowadays, the Web 2.0 websites, including YouTube [115], Facebook [37], Twitter [103], and others, are so popular that they have drastically changed the content distribution landscape, and have become one of the most important parts in people's everyday life.

The characteristics of Web 2.0 are considered as the key factors to the success of these websites. They include:

Rich user experience. The websites combines GUI-style (Graphical User Interface) applications and multimedia contents. A web-based software provide users a similar experience to a computer-based software [97].

Collective intelligence. The base knowledge of general users of the web is contributing to the content of the web [97]. Users have become the content providers and therefore the contents are dynamic.

Social network. A social network is a social structure made up a set of individuals and ties between them [95]. The web is founded on the idea of hyperlinking, that is, contents are linked to each other, creating an organic growth of connections and activities.

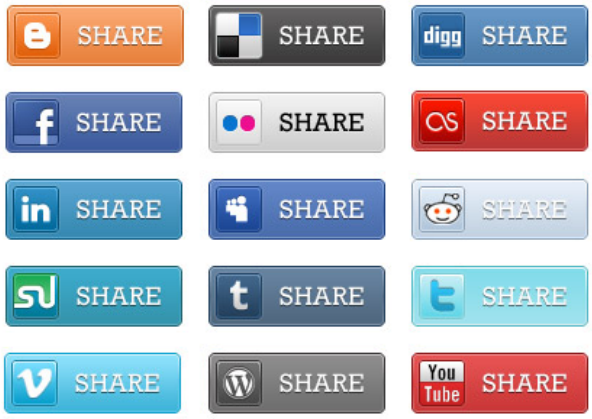


Figure 1.1: Share buttons of social media websites (originally from Wikipedia)

Social media, a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, allow the creation and exchange of user-generated contents [60]. Enabled by ubiquitously accessible and scalable communication techniques, social media has substantially changed the way organizations, communities, and individuals communicate [63]. There are many types of social media services, and this thesis is centred on the two most important types, namely, *video sharing service* and *social networking service*. Other notable social media services include social news (e.g., reddit [86]), social bookmarking (e.g., Delicious [35]), location-based service (e.g., Foursquare [43]), question-and-answer (e.g., Quora [85]), encyclopedia (e.g., Wikipedia [110]), etc.

In social media, the contents, as well as the users, have become inter-connected. One of the most important features of social media is *sharing* (refer to Figure 1.1). Users can share their statuses including feelings, activities, and location information, as well as resources, blogs, photos, and videos.

In this chapter, we first present the background of video sharing service and social networking service. Then we present the motivations of this thesis, and summarize the contributions. Finally we describe the organization of the thesis.

1.1 Background

We present the background of the two services, introducing three representative social media applications.

1.1.1 Video Sharing Service

In the traditional video on-demand and live streaming services, videos are provided by the servers and are streaming to the users proactively. A number of new generation video sharing websites, represented by YouTube, utilize Web 2.0 techniques to provide users opportunity to make their videos known to others, by such simple operations as embedding and sharing.

Established in 2005, YouTube is no doubt the most significant and successful video sharing website. YouTube allows registered users to upload videos, mostly short video. Users can watch, embed, share, and engage with videos easily. As one of the fastest growing websites in the Internet, YouTube once served 100 million videos per day in 2006 [51], yet this number had grown to 1 billion, 2 billion, and 3 billion in 2009 [118], 2010 [119], and 2011 [120], respectively. As of January 2012, YouTube has 4 billion daily views, and every second, one hour of video is uploaded on YouTube by users around the world [121]. It is reported that in 2011, YouTube had almost 140 views for every person on earth [125]. YouTube is ranked third of the top traffic sites by Alexa [3], as over 800 million unique users visit YouTube each month and over 3 billioin hours of video are watched each month [125]. The success of similar sites (e.g., Dailymotion [33], Metacafe [69], Vimeo [106], Youku [114], and Tudou [102]), and the blockbuster acquisition of YouTube by Google [52], further confirm the mass market interest of video sharing services.

For social media content distribution, video data is arguably much more important than the other types of data. Video data have much larger size, comparing with texts and pictures. Industry predicts that 90% of all Internet traffic will be video in the next three years [78]. Moreover, people are spending more and more time watching online videos nowadays, as a survey revealed that Americans spend 3.5 hours per week watching online videos nowadays [49], accessing video contents requires higher quality of service (QoS), which directly impact the development of the video sharing service.

1.1.2 Social Networking Service

Social networking services provide an online platform to connect people with social relations, e.g., friends, classmates, colleagues in the real world, or people who share common interests.

Facebook, founded by Mark Zuckerberg in 2004, is the current number one social networking website on the Internet. Facebook had 901 million monthly and 526 million daily active users as of March 2012 [39], and is ranked second of the top traffic sites by Alexa.

Facebook provides users a platform to connect with friends, by updating status, uploading photos, sharing videos, commenting and "liking" other's posts, etc. Currently there are 3.2 billion of likes and comments per day, and 300 million of photos are uploaded per day [31]. Facebook opens its API for developers to build thousands of applications and games, which makes Facebook more enjoyable. A social media report in the third quarter of 2011 revealed that Americans spent 53.5 billion minutes, that is, over 101 thousand years, on Facebook monthly [94]. Facebook's recent IPO (Initial Public Offering) filing [38] further confirms the significant impact of this service as a social media.

Another important social networking website, Twitter, is the representative of microblog, a simpler but much faster version of blog. It allows users to send text-based posts, called "tweets", of up to 140 characters. Although short, the tweets can link to richer contents such as images and videos. By following friends or interested accounts, such as news presses, celebrities, brands, and organizations, users can get the real-time notification, and can spread the post by "retweet". Since its establishment in 2006, Twitter has been growing rapidly, now with 200 million users [92] generating over 200 million tweets per day as of 2011 [104]. Like Facebook, Twitter also opens its APIs so that there are over 150 thousand registered Twitter applications [104], among which a large portion are for mobile devices, making Twitter easier to access.

By sharing contents in the social networking service, a "word-of-mouth" communication emerges. In this thesis, we are in particularly interested in video sharing in social networks. As the integration of video sharing and social networking, videos now become easier to access. This feature has not only significantly changed the paradigm of content distribution and spreading, but also brought unprecedented challenges to network traffic engineering.

1.2 Motivation

Understanding the features of social media websites is crucial to network traffic engineering and to sustainable development of this new generation of services. In the literature, video sharing websites, especially YouTube, had been extensively studied [13, 41, 42, 50, 72, 127], including our previous works [15, 16, 17, 19, 21, 24, 25, 26, 28]. Social networking websites had also been studied extensively [7, 59, 64, 91, 98]. However, there are still areas that have not been explored, and there also have been several significant changes on the social media websites over the past few years:

- YouTube displays advertisements on the webpages to monetize videos, which has been the main source of YouTube's revenue. It is reported that YouTube monetizes over 3 billion video views per week globally [125]. YouTube further introduced the *YouTube Partner Program* [116], so that content owners whose copyrighted videos and channels have pulled a large audience can earn income from advertisements revenues. There are over 30 thousand partners around the world and YouTube has paid out millions of dollars a year to them [125]. The introduction of YouTube partners has essentially changed the content landscape. YouTube once is a typical user-generated content service, where videos are uploaded by ordinary users. Now, companies and organizations such as EA, ESPN, Machinima, Warner Brothers, have become YouTube partners and are providing premium videos on YouTube. The YouTube partners have largely improved the quality of YouTube videos, and have further increased YouTube's popularity. The understanding of YouTube Insight is of great potential value to the YouTube partners.

- Video sharing services and social networking services have become highly integrated. YouTube enables automatic post on social networking websites such as Facebook and Twitter based on their preferences, and users can also share interesting videos on their social networking website pages. In fact, YouTube has been brought directly into Google+ [53], making it easier to watch and share [77]. Statistics show that 500 years of YouTube video are watched every day on Facebook, and over 700 YouTube videos are shared on Twitter each minute; the YouTube player is embedded across tens of millions of websites [125]. The videos are spreading in the social networks, bringing significant challenges not only to the social network management, but also to the network traffic engineering.

- With the pervasive penetration of wireless mobile networks, the advanced development of smartphones and tablets, and massive market of mobile applications, social media contents can be generated and accessed at any time and anywhere easily. It is reported that YouTube mobile gets over 600 million views a day, and traffic from mobile devices tripled in 2011 [125]. Furthermore, mobile devices are predicted to dominate Internet access in the near future [100]. For video sharing, the new video creation, deployment, and spreading trend, beyond conventional media, has brought up numerous well-known Internet memes [101] and celebrities such as Justin Bieber [124].

- As the social media services are developing at a great pace, scalability and provision of resources will eventually become an issue. The huge demands of bandwidth and storage for social media contents has brought great challenges to the network engineering. The recently emerged cloud service solves the dilemma for developers at the early stage of the service, and a migration to cloud is essential as well. However, the existing research works focus on preserving the social relationship only, while an important factor, user access pattern, is largely overlooked, which can cause unbalance problem in the cloud scenario.

Therefore, this thesis tries to further understand the characteristics of video sharing service and social networking service, and leverages social media content distribution, providing better quality of service.

1.3 Contributions

In this thesis, we conduct four joint and extensive studies on social media content distribution. In particular, we make the following contributions:

- We for the first time analyze a large-scale of YouTube *Insight* data, which are provided by YouTube analytics system. We reveal a number of unique features of YouTube partners. Specifically, we further investigate the video popularity, and characterize viewing surges, daily pattern, visiting behaviour, as well as analyze the impact of user subscription and engagement on video view. We also reveal the breakdown of referral sources, and specifically investigate the impact of video suggestion and external website referral on video views. On one hand, our study extends the existing research works, revealing more interesting features of video sharing service, and thus has great contribution to the literature. On the other hand, our study is of great potential value to the YouTube partners, providing useful suggestions on the strategy for YouTube partners to adapt their content deployment and user engagement, so as to attract more subscribers, generate more video views, and subsequently increase their revenues. This work is under review for publication [18].

- We investigate the characteristics of video spreading in social networks, based on large-scale data traces from Renren, the largest Facebook-like social networking service in

China. We examine the user behaviour from diverse aspects, and identify different types of users in video propagation and evaluate their activities. We also examine the video link propagation patterns, in particular, the temporal distribution during propagation as well as the typical propagation structures, and reveal more details beyond stationary coverage. We further extend the conventional epidemic models to accommodate the diversity of the propagation, and our model effectively captures the propagation process of video sharing in the social networks, serving as a valuable tool for such applications as workload synthesis, traffic prediction, and resource provisioning. This work is under review for publication [20].

- We have conducted a user questionnaire survey to directly understand user preference and social interest in online video sharing. The survey result reveals an interesting coexistence of live streaming and storage sharing, and the users are generally more interested in watching their friend's videos. More importantly, our survey reveals that many of the users are willing to share their resource to assist others with close relations, implying node collaboration is a rationale choice in this scenario. Therefore, we propose COOLS (Coordinated Live Streaming and Storage Sharing), a system for efficient peer-to-peer posting of user-generated videos. Through a novel ID code design that embeds nodes' locations in an overlay, COOLS leverages stable storage users and yet inherently prioritizes live streaming flows with short startup delay. We further improve the design to achieve better performance. This work was published as a conference paper [22] and a journal paper [27].

- We first reveal that one important factor, user access pattern, is overlooked in the existing works on partitioning social networks. By examining a large collection of YouTube video data and Twitter user data, we demonstrate that partitioning the social graph entirely based on social relationship will lead to unbalanced partitions in terms of access. We further analyze the role of social relationship in different social media services, taking three representatives as examples. We conclude that user access pattern should be taken into account and social relationship should be dynamically preserved. We formulate the problem as a k-medoids clustering problem with constraint, and propose a novel Weighted Partitioning Around Medoids (wPAM) solution. We also extend our study from client/server scenario to peer-to-peer scenario. Part of this work was published as a conference paper [23].

The four studies in this thesis are highly related. Firstly, based on large-scale Insight data of YouTube partners, we extend the existing studies on YouTube, revealing new observations. Specifically, we reveal the traffic source of referral views, and study the impact of external websites on YouTube video views. This integration of video sharing and social networking is further investigated in the study on video spreading in social networks. The integration of video sharing and social networking motivate us the new scenario of coexistence of sharing and streaming, and thus we propose new system to achieve better quality of service. Lastly, as the development of social media service, scalability and provision of resources will eventually become an issue, and cloud is a suggested solution. In particular, evidently from the above studies, an important factor, user access pattern, is taken into account in our study on migration of social media.

1.4 Organization of the Thesis

The remainder of the thesis is structured as follows:

- In Chapter 2, we review the related works in the literature, and in particular revisit our previous studies.
- In Chapter 3, we investigate Insight data of YouTube partners. We reveal new features of YouTube and provide helpful guidance to YouTube partners.
- In Chapter 4, we investigate video spreading in social networks. We study user behaviour as well as the characteristics of spreading structure. We use an extended epidemic model to describe the video spreading.
- In Chapter 5, based on our measurement study and a user questionnaire survey, we propose a novel system, COOLS, in the new scenario of coexistence of live streaming and storage sharing.
- In Chapter 6, we investigate the migration problem for social media to cloud service. We take user access pattern into account, formulate the problem as a k-medoid problem with constraint, and solve the problem by extending the classic PAM method.
- In Chapter 7, we conclude the thesis, and also discuss some future works.

Chapter 2

Related Works

In this chapter, we first revisit our previous studies. Then we review the measurement studies on video sharing and social networking services, as well as other related works, namely, information propagation, online video streaming, and cloud computing. We also present some basic statistical and mathematical concept and technique utilized in this thesis.

2.1 Previous Studies on Video Sharing

2.1.1 Understanding Statistics of YouTube

To understand the features of YouTube, we presented an in-depth and systematic measurement study on the characteristics of YouTube, based on over 5 million video data crawled in a 1.5-year span [15, 16, 17, 19, 25].

The most distinguished difference of YouTube from the traditional media contents is the video length. As shown in Figure 2.1, 98.0% of the video lengths are within 600 seconds, which is mainly due to the limit of 10 minutes imposed by YouTube on regular users uploads.

The rank distribution of the number of views was examined. As shown in Figure 2.2, though the plot has a long tail on the linear scale, it does not follow the well-known Zipf distribution $y = x^{-a}$ [128], which is a straight line on a log-log scale. The tail drops tremendously, indicating that there are not so many unpopular videos as Zipf's law predicts. We found that the Gamma [46] and Weibull [109] distributions both fit better than the Zipf (refer to Figure 2.2). We also investigated the correlation between video's age and number of views, finding that different videos have various *growth trends*.

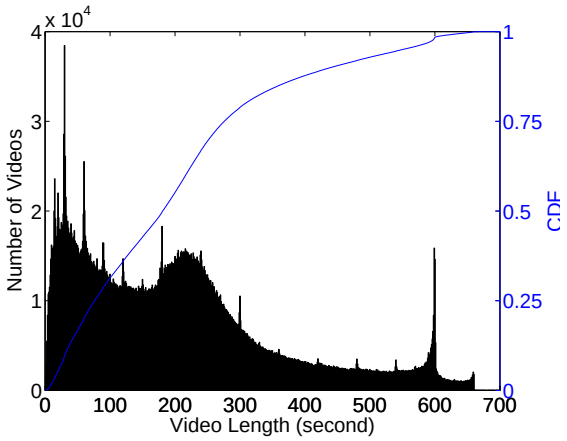


Figure 2.1: Distribution of YouTube video length [25]

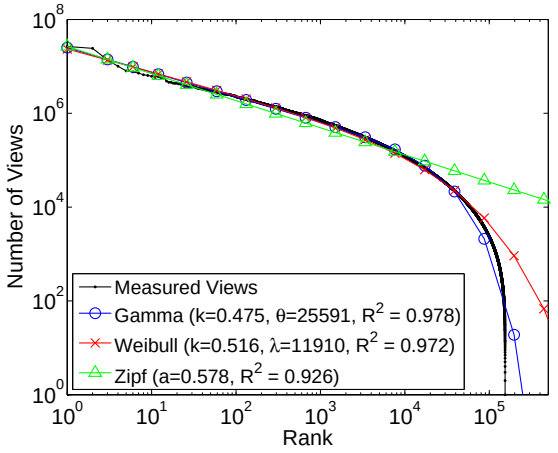


Figure 2.2: YouTube videos' views against rank [25]

The growth trend was then studied. We used a power law

$$v(x) = v_0 \times \frac{(x+\mu)^p}{\mu^p}$$

to model growth trend p, where μ is the number of weeks that the video has been uploaded before the data collection, and v_0 is the number of views in the first dataset crawled. As a result, over 80% of the videos have factors less than 1, indicating that most videos grow more and more slowly as time passes. We further studied the *active life span*, which is defined as the time when the number of views increases by a factor less than a threshold t from the previous week. Therefore, the active life span l follows $\frac{v(l)}{v(l-1)} - 1 = t$, and we had

$$l = \frac{1}{\sqrt[p]{1+t}-1} + 1 - \mu.$$

The active life provides a way to estimate the temporal locality, and we found that most of the videos have been watched frequently only in a short span of time, and after a video's active life span, fewer people will access it. This characteristic has good implications for web caching and server storage.

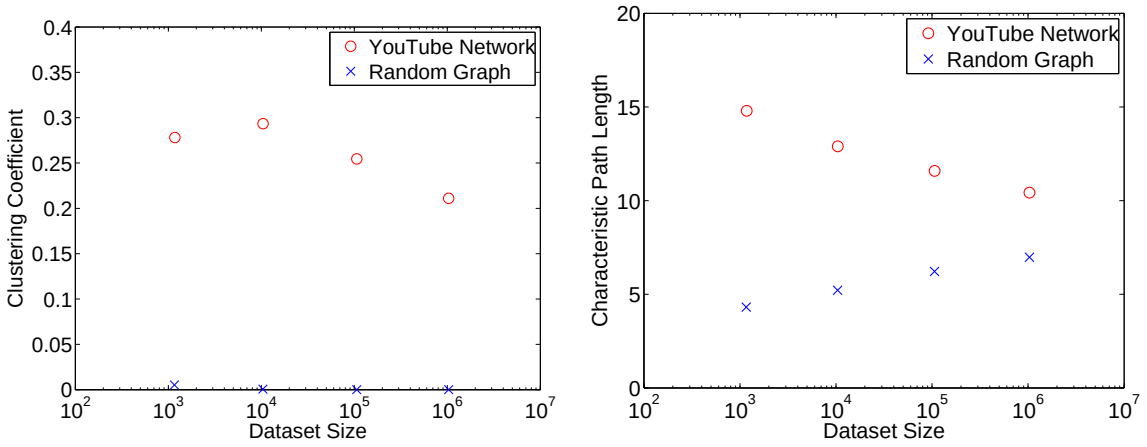
We also studied the social networks in YouTube. We measured the graph topology for the network of YouTube videos, by using the related links in YouTube pages to form directed edges in a video graph. The visual illustration for part of the network (about 4000 nodes) is shown in Figure 2.3. From the entire crawled data, we obtain four datasets for measurement, each consisting different order of magnitude number of videos. For comparison, we also

Figure 2.3: Sample graph of YouTube videos and their links [25]

generate random graphs that are strongly connected, and each has the same number of nodes and average node degree of the datasets. Figure 2.4a shows the clustering coefficient for the graph. The clustering coefficient is quite high (between 0.2 and 0.3), comparing with the random graphs (nearly 0). Figure 2.4b shows the characteristic path length for the graphs. The average diameter (between 10 and 15) is only slightly larger than the diameter of a random graph (between 4 and 8), which is quite good considering the still large clustering coefficient of these datasets. This phenomena verifies that the YouTube graph is a small-world network, which can be explored to enhance YouTube service.

2.1.2 Enhancing Short Video Sharing

The sustainable development of YouTube and other video sharing sites is severely hindered by the intrinsic limit of their client/server architecture. A shift to the peer-to-peer paradigm has been widely suggested with success already shown in live video streaming and movie-on-demand. Unfortunately, our measurement demonstrates that short video clips exhibit drastically different statistics, which would simply render these existing solutions suboptimal, if not entirely inapplicable. On the other hand, our measurement reveals interesting social networks with strong correlation among the videos. We therefore presented NetTube,



(a) Clustering coefficient (b) Characteristic path length

Figure 2.4: Small world characteristics of YouTube videos [25]

a novel peer-to-peer assisted delivery framework that explores the user interest correlation for short video sharing [21, 24, 26, 28].

In NetTube, the server stores all the videos and supplies them to clients. The clients also share the videos through peer-to-peer communications. A distinct feature of NetTube is that a peer caches all its previously played videos, and makes them available for re-distributing. As such, for a client interested in a particular video, all the peers that have previously downloaded this video can serve as potential suppliers, forming an overlay for this video, together with the peers that are downloading this video.

In particular, since a client in general will watch a series of videos, it is necessary to quickly locate the potential suppliers for the next video and enable a smooth transition. To this end, NetTube introduces an upper-layer overlay on top of the overlays of individual videos. In the upper-layer overlay, given a peer, neighbourhood relations are established among all the overlays that contains this peer. This is a conceptual relation that will not be used for data delivery; instead it enables quick search for video suppliers in the social network context. Figure 2.5 shows this bi-layer overlay.

In the system, each NetTube peer needs to maintain a record of its cached videos, and makes it available for searching. The server also needs to keep track of peers' videos. Hence we proposed a space-efficient Bloom filter index. We represent a series of video IDs, by a Bloom filter [9]. Each time a peer finishes downloading a video, the video ID is mapped to update the Bloom filter index.

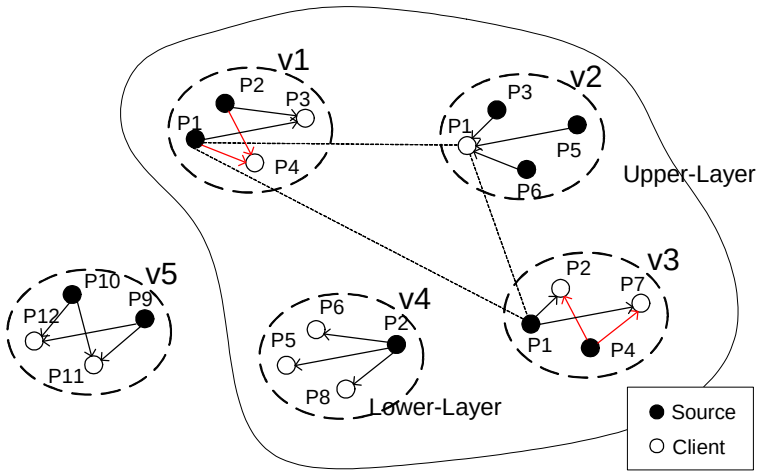


Figure 2.5: Illustration of a bi-layer overlay [24]

The system adopts a mesh-based overlay, where the peers pull expected data from a set of partners (other peers or the server) through a sliding-window-based scheduling algorithm. Yet, given the shorter video length, the startup and playback delay would be amplified from the perception of users. Given the users are less patient in waiting for short videos, more trials of joining/leaving would occur, leading to higher churn rates. To address these challenges, we proposed a novel delay-aware scheduling that is customized for the short videos. Specifically, we implement an intelligent indicator in the downloading buffer to tell whether the peer is about to encounter delay. If yes, we will utilize an aggressive strategy for transmitting the data to mitigate delay. That is, the senders will prioritize such requests, even if they have to suspend some other transmissions.

To achieve fast and smooth transition, we further introduced a cluster-aware pre-fetching in NetTube, where the system pre-fetches *video prefixes* during the playback of the current video. With the existence of video interest correlation, the hit-rate of pre-fetching can be very high after a client plays back multiple videos. The intuition is that, given that the videos are generally clustered, the videos pre-fetched for the current video are likely in the related list of the next video, and so forth.

We performed extensive simulations and prototype experiments on PlanetLab [83] to evaluate the performance of NetTube. The results shown that NetTube greatly reduces the server workload, improves the quality of playback, and scales well.

2.2 Studies on Video Sharing Services

There are numerous studies investigating video sharing services, especially YouTube. Cha *et al.* [13] presented measurement results that are crucial in understanding YouTube-like systems and provide valuable implications. Compared with non-UGC services, the authors found that content production is significantly faster with less effort, because UGC video length is shorter by two orders of magnitude. They also found that 10% top popular videos account for nearly 80% of the views, indicating that YouTube is highly skewed towards popular videos. This finding implies that caching can get high hit ratio since only a small portion of videos will be requested frequently. The authors also discussed peer-to-peer possibility: since the video requests are highly skewed, determined by a request rate, peer-to-peer can benefit the system. The trace-driven analysis has shown that the server workload can be reduced by 41% when users share only videos while they are online (28 minutes).

Mislove *et al.* [72] investigated four social networks including YouTube. They observed that all of the networks show power-law distribution in the node's degree, and also found the high correlation between out and in degree. Moreover, the average path lengths of the four social networks are calculated as between 4 and 6, much smaller than that of the web graph. They further found the high correlation coefficient of the social networks, comparing with web graph. The observations indicate that these social networks are small world.

Gill *et al.* [50] presented a traffic characterization study on YouTube. The authors implemented a YouTube traffic monitor on a campus network to collect traces. They characterized YouTube video files in terms of size, duration, bit-rate, etc. They further investigated locality characteristics of YouTube videos accesses. In particular, only 10% of the video are watched again on the following day. The top popular videos do not contribute much to the total videos viewed on campus on a daily basis, probably because users are directed to the videos shared by friends, e.g., on Facebook, instead of browsing the most viewed list.

Finamore *et al.* [42] also investigated the traffic usage of YouTube, focusing on mobile users. They found that users tend to abort the playback very soon, with 60% of videos being watched for less than 20% of their duration, and PC and mobile users show the same result. Because of this user behaviour, the authors demonstrated that a great amount of downloaded data are wasted, bringing a critical issue to both ISP and YouTube CDN.

Zhou *et al.* [127] specifically investigated the impact of YouTube recommendation system. They found that search and related video are the two top important view sources,

and there is a strong correlation between the number of views of a video and that of its top referrer video. They also found that YouTube recommendation provides more diversity on video views, implying that YouTube recommendation helps viewers discover more videos of their interest rather than the popular videos only.

Figueiredo *et al.* [41] specifically characterized the growth patterns of YouTube videos, based on the Insight statistics displayed on YouTube video webpage [127]. They categorized the videos into three datasets, namely, top videos, removed pirate videos, and random videos. They investigated the types of the referrers on three datasets, finding that copyrighted videos tend to get most of the views earlier, while videos in the top lists tend to experience sudden bursts of popularity. They also shown that YouTube internal mechanisms play important roles on the video popularity.

As we discussed, there have been several major changes on YouTube and other video sharing services. Therefore, understanding the new features such as YouTube partners is crucial to its service. Moreover, there are also some important characteristics of YouTube that have not been studied, such as viewing surge, impact of external sources, and so on. Therefore in this thesis, we will further investigate the unique features of YouTube.

2.3 Studies on Social Networking Services

Online social networking services have also been extensively studied in the literature. Kwak *et al.* [64] crawled the entire Twitter site, including 41.7 million user profiles, 1.47 billion relations, $4,262$ trending topics and 106 million tweets. One interesting finding is that Twitter users are separated by average degree of 4.12, which is surprisingly short, given that 77.9% of user pairs are connected one-way. The authors compared popular topic in Twitter with other media, confirming that Twitter is an effective media for breaking news, and a good portion of users have participated in trending topics. They also investigated the impact of retweet (spread other's post), which largely expands the range of the information propagation. The size and height of retweet diffusion trees follow power-law distribution, and half of the retweets occur within an hour.

User behaviour in social networking services is also investigated. Benevenuto *et al.* [7] characterized user behaviour from a social network aggregator that accessed to four popular sites. They found that users spend various time on social networking site, 51% are less than 10 minute over 12 days, yet 14% are more than 1 hour. The authors proposed a clickstream

model to describe user behaviours in social networking service, which can be utilized to improve personalized web interface. The authors further observed that users interact with friends as well as users that are two or more hops away, which has great impact on the information and content propagation. In another work, Schneider *et al.* [91] reconstructed social networking site clickstreams from traces obtained from different ISPs. They found that users tend to stay within the same activities, and the sessions are typically longer than 30 minutes, but continuous interactions are rare for sessions longer than 10 minutes.

The above two studies are however constrained by the time range and website structure. Thus latent interaction is another important aspect to understand user behaviours, for example, which visitors have browsed a user's page. Jiang *et al.* [59] crawled a campus network of Renren and reconstructed visitor histories. They found that most users do not visit in return; yet, compared to strangers, friends have relatively higher probability of reciprocal visits. Comparing latent and visible interactions, the authors found that users are more active in viewing profiles than leaving comments, and consequentially, latent interactions cover a wider range of friends than visible interactions.

In this thesis, we will further investigate social networking service, focusing on the video spreading in the social networks, which is crucial to the social network management, as well as the network traffic engineering.

2.4 Studies on Information Propagation

There have been significant studies on information propagation in different types of networks. Wang *et al.* [107] examined an email communication dataset, and found that the information propagation depends on the social and organizational context, and the structure of the propagation can be captured by a stochastic branching model. Dyagilev *et al.* [36] introduced Discrete Gaussian Exponential (DGX) model for mobile network data exchanges. However, these works do not focus on social networking services, although certain social relations implicitly exist in email and mobile phone communication networks.

Traditionally, Internet users obtain information by searching or browsing portal websites. With the emergence of social media, "word-of-mouth" becomes a popular way for information discovery and propagation. Rodrigues *et al.* [89] investigated the URL propagation in Twitter, in which the propagation can reach to a large range of users. Interestingly, domains whose URLs are spread widely in Twitter tend to be different from domains that

are popular on the web, indicating that word-of-mouth gives all contents a chance to be propagated to a large audience. The information propagation tree for URL are wide and shallow, which is in sharp contrast to the narrow and deep trees in Internet chain letters.

Most of the existing works on information propagation in social networks have largely targeted on viral marketing, with an objective of maximizing the information coverage. Tang *et al.* [98] investigated multiple relationships among users in social networks. They proposed a random walk-based algorithm for relationship classification, which facilitates advertising specific products. Budak *et al.* [12] presented an Independent Cascade Model, and examined an eventual influence limitation problem. It limits "bad" information propagation by maximizing "good" information propagation. An earlier work by Cha *et al.* [14] studied Flickr, one of the most popular photo sharing sites. They showed that information propagation in Flickr is dominated by social links, but is limited to users that are close to the uploaders. They also found that the propagation takes a long time with certain locality.

In this thesis, we will show that, despite certain similarities, video sharing in social networks possesses quite different characteristics, and we focus on the dynamics of the spreading process instead of the final stationary coverage, which is critical towards understanding video sharing as well as the impact of social media.

2.5 Studies on Online Video Streaming

Numerous peer-to-peer protocols have been developed for live or on-demand video streaming, which can be broadly classified into two categories according to their overlay structures [67], namely, tree-based overlay, represented by Chunkyspread [105], and mesh-based overlay, represented by CoolStreaming [126]. In tree-based system, streaming data are pushed to children peers, while in mesh-based system, peers need to seek partners, exchange the information of data availability, and pull the streaming data. Therefore, tree-based system is much more efficient than mesh-based system; the delay is short and the overhead is low. However, mesh-based system is resilient to dynamics, while in tree-based system, handling node dynamics is expensive. To achieve both efficiency and resilience, hybrid overlays are proposed, such as Labeled Tree [108].

On the other hand, we have seen earlier attempts toward joint live and on-demand peer-to-peer streaming. For example, BitTorrent has enabled a streaming mode, so that user could watch on-demand video while downloading it [8]; also, peer-to-peer streaming

platforms such as PPLive now provide both live and VoD modes [57].

However, efficient implementation and more importantly, seamless integration of the video sharing and social networking, remains a great challenge. Therefore in this thesis, we will examine a new scenario that video streaming and storage sharing coexist, and propose a novel system integrating the two types of users, achieving better system performance.

2.6 Studies on Migration to Cloud

In order to move social media contents to cloud, one important step is to partition the social graph. There are some works trying to detect the communities and partition social graph. Newman *et al.* [75] studied a set of algorithms for discovering community structure. The algorithms iteratively remove edges, identified by "betweenness" measure, from the network to split it into communities. Mishra *et al.* [71] introduced a new criterion that overcomes the limitations that clusters typically do not overlap, by combining internal density with external sparsity in a natural way. SNAP [6] is a tool for analyzing and partitioning small-world network by maximizing the modularity of the graph. SPAR [84] considers the cloud scenario, replicates linked nodes in the same server, and tries to minimize the replications.

The above mechanisms are effective on preserving the social relationship. However, as mentioned, while the social relationship is an important factor to the efficiency of the cloud computing, user access pattern is overlooked in their works. Therefore in this thesis, we will take this factor into account to achieve load balance.

2.7 Miscellaneous

2.7.1 Distribution

In this thesis, we will extensively examine the distribution of various metrics. We mainly utilize three types of representations:

Rank distribution shows the values of a metric in descending order, and thus the x-axis represents the rank. For example, Figure 2.2 shows the rank distribution of YouTube video views.

***Probability density function* (PDF)** shows the probability of a metric being a specific value. The x-axis represents the value and the y-axis represents the probability. It

is called probability mass function (PMF) for discrete variables. Also, we sometimes show the count instead of the probability for y-axis.

***Cumulative distribution function* (CDF)** shows the probability of a metric being less than and equal to a specific value x. The x-axis represents the value and the y-axis represents the probability; the probability increases from 0 at the minimum x to 1 at the maximum x. For example, CDF of YouTube video lengths is shown in Figure 2.1.

Different distribution functions are used to fit the above curves. In this theses, we mainly use the following distributions:

***Zipf's law* [128]** is used to fit the rank distribution. It is often plotted in log-log scale, and exhibits a straight line. The function is $f(x) = \frac{C}{x^a}$, where a determine the slope of the line and C is the value of the top-one item.

***Pareto* [81]** , as well as the Zipf's law, is in the family of Power law distribution. It also exhibits a straight line in log-log scale plot. The PDF is $f(x) = \frac{\alpha x_m^\alpha}{x^{\alpha+1}}$ for $x \geq x_m$. There is another form of Pareto distribution, called *generalized Pareto distribution* (GPD). Its CDF is $f(x) = 1 - (1 + \xi \cdot \frac{x-\mu}{\sigma})^{-\frac{1}{\xi}}$, where μ determines the location, σ determines the scale, and ξ determines the shape of the curve.

***Weibull* [109]** is another important distribution. Its PDF is

$$f(x) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0 \\ 0 & x < 0 \end{cases},$$

where λ determines the scale and k determines the shape of the curve; its CDF is $1 - e^{-(x/\lambda)^k}$.

To determine the goodness of fit on the distribution, *coefficient of determination* R^2 is often used to describe how well it fits a set of observations. R^2 is defined as

$$1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2},$$

where f are generated data or modelled values, y are the real data and $\bar{y}$ is the mean of the real data [30].

2.7.2 Correlation

In this thesis, we will compute the correlation between two metrics in several circumstances. Specifically, we calculate the *Pearson correlation coefficient*, referred to as CC. The CC is calculated as

$$\frac{1}{n-1}\sum_{i=1}^{n}\Big(\frac{X_i-\bar{X}}{s_X}\Big)\Big(\frac{Y_i-\bar{Y}}{s_Y}\Big),$$

where X and Y are two random variables, $\bar{X}$ and s_X are the mean and standard deviation of X. The CC is between -1 and 1, as being 1 indicates the perfect positive linear relationship and being -1 indicates the perfect negative linear relationship, and approaching 0 indicates the decrease of relationship [82].

Chapter 3

Insight Data of YouTube: From a Partner's View

YouTube is arguably the most popular online video sharing site nowadays. To further augment its service with better revenue, it has started working with content owners with a large audience (known as *YouTube partners*). By uploading high-quality premium videos, the partners have essentially changed user-generated content feature of YouTube. Understanding the latest YouTube access pattern is thus crucial to both YouTube and its partners, as well as to other providers of relevant services. In this chapter, we for the first time analyze a large-scale YouTube dataset from a partner's view. We make effective use of *Insight*, a new analytics service of YouTube that offers inside statistics for partners about their content accesses and audience behaviour. From the raw Insight data that are confined to simple scalars and charts, we reveal the inherent relationship among various metrics that affect the popularity of the videos. Our findings facilitate YouTube partners to adapt their content deployment and user engagement strategies, having great potentials for them to collaborate with YouTube to generate more views and subsequently increasing their revenues.

The chapter is organized as follows. The background is introduced in Section 3.1. We present the data collection methodology and dataset description in Section 3.2. Section 3.3 and Section 3.4 analyze the characteristics of two reports from the Insight data, respectively. We conclude our findings in Section 3.5.

3.1 Introduction

Over the past few years, YouTube has become the most popular online video sharing website. It allows users to easily watch, embed, upload, share and interact with videos. YouTube displays advertisements next to the video players on the webpages to monetize video, which has been the main source of YouTube's revenue. It is reported that 3 billion video views are being monetized per week globally [125]. In 2007, it introduced the YouTube Partner Program [116], so that content owners whose copyrighted videos and channels have pulled a large audience can earn income from the advertisements revenues. YouTube now has over 30 thousand partners and pays out millions of dollars a year for them [125].

As a typical user-generated content service, YouTube had substantially changed the traditional online video service. The introduction of YouTube partners however has essentially brought a new content landscape. Companies and organizations such as EA, ESPN, Warner Brothers, now have become YouTube's important partners and are providing high-quality videos on YouTube, attracting more and more audience. Over the past few years, more and more small businesses and individuals have also partnered with YouTube to benefit from the monetization of their videos. Hundreds of partners are making six figures a year, and partner revenue has doubled for four years in a row [125]. Google also invests in Machinima [96], one of the most popular networks on YouTube, to produce more appealing videos, further implying the importance of YouTube partners. By uploading premium videos, YouTube partners have further increased YouTube's popularity. As of January 2012, YouTube has 4 billion daily views, and every second, one hour of video is uploaded on YouTube by users around the world [121]. Therefore, understanding the latest YouTube access patterns is crucial to both YouTube and its partners.

The statistics of videos are of great potential value to the YouTube partners. For example, which videos are popular? which external websites are referring more views? The partners can leverage these statistics to adapt their content deployment and user engagement strategies. To help the YouTube partners with this goal, YouTube introduced a new service, YouTube Insight Analytics [117], providing various basic statistics of videos and channels. Unfortunately, the current Insight Analytics service only provides simple scalars and charts of metrics, and fails to provide more interesting and complicated implications. Such information can affect partners' user engagement strategy and also can be very important for content deployment purposes. Based on these arguments, more advanced analysis

of Insight data is helpful for YouTube partners to better achieve their goals.

In the literature, video sharing services, especially YouTube, have been studied extensively [13, 17, 50, 72]. They found a number of distinguished features of YouTube from the traditional VoD system. For example, video lengths are short, user access pattern has unique characteristic, and social networks exist in YouTube. There are also some works specifically studied the impact of related videos and referrers on video views [41, 127]. To the best of our knowledge, the Insight data were not available in the existing research studies, since they are private to YouTube partners. Therefore, revealing more unique features of YouTube to extend the existing studies has great contribution to researchers.

In this chapter, we conduct a measurement study on analyzing two large-scale Insight datasets of YouTube partners. We reveal a number of unique features of YouTube partners. Specifically, we further investigate the video popularity, and characterize viewing surges, daily pattern, visiting behaviour, as well as analyze the impact of user subscription and engagement on video views. We also reveal the breakdown of referral sources, and specifically investigate the impact of video suggestion and external website source on video views.

3.2 Data Collection

YouTube provides an Insight Analytics dashboard on the webpage, showing some simple scalars and charts. Figure 3.1 gives an example of the dashboard. To further investigate, we implemented a crawler to collect the raw Insight data using YouTube Data API [123]. Because Insight data are only available to the content owner, the crawler was authenticated to collect such data. The Insight data were collected from the date when each channel was established to February 29, 2012.[1]

The raw Insight data include four datasets. In this work, we investigated two of them. An brief explanation of the two Insight datasets is as follows:

- in views datasets, each trace records the number of views and other statistics such as unique visitors, likes, comments, and subscriptions, for a video in a specific region, on a specific day. Below we have listed the title and some example views traces:
  `Date,Region,VideoID,Title,Views,Unique users,Unique users (7 days),`

[1]Due to the limitation of YouTube Insight service, the earliest date that can be crawled is March 1, 2009. Nevertheless, our collected data range up to 3 years.

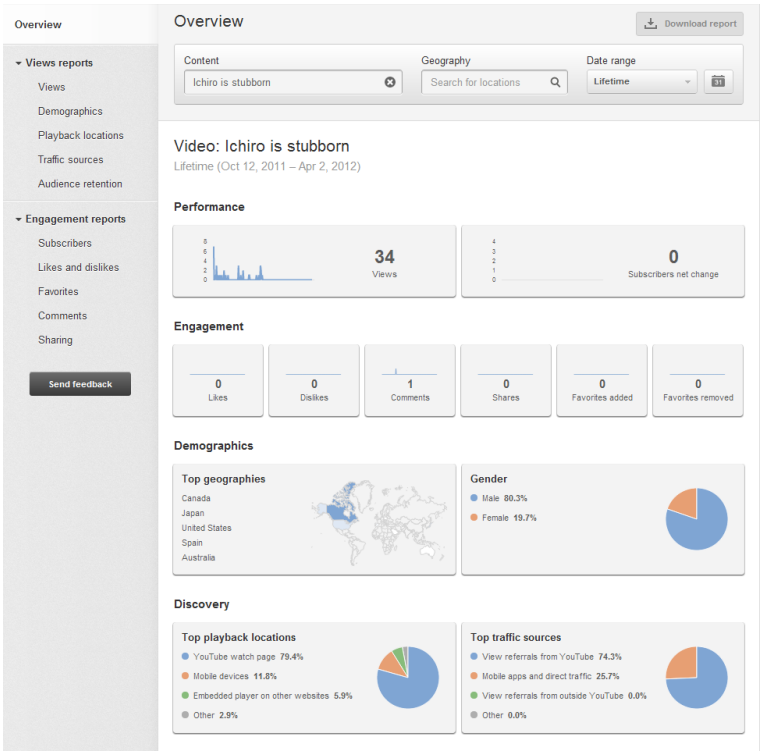


Figure 3.1: YouTube Insight dashboard

```
Unique users (30 days),Popularity,Comments,Favourites,Rating 1,
Rating 2,Rating 3,Rating 4,Rating 5
2011-10-12,JP,B15e_P2QaTc,Ichiro is stubborn,4,2,2,2,0.000809,0,0,0,
0,0,0,2
2011-10-12,CA,B15e_P2QaTc,Ichiro is stubborn,2,2,3,3,0.000632,1,0,0,
0,0,0,1
```

- in referrers datasets, each trace shows how users have reached a video in a specific region, on a specific day. It provides the number of views from a particular source, including the related video list, search results, external websites, etc. Below we have listed the title and some example referrers traces:

```
Date,Region,VideoID,Title,Source type,Detail,Referred views
2011-10-12,JP,B15e_P2QaTc,Ichiro is stubborn,YT_SEARCH,shiba,1
2011-10-12,JP,B15e_P2QaTc,Ichiro is stubborn,YT_SEARCH,shiba inu,1
```

Table 3.1: Summary of Insight Data

	Channel A	Channel B	Channel C
Genre	movie trailers	game trailers	game playing videos
Months since established	> 36	> 36	12
# of videos	$4,879$	$1,975$	$1,051$
# of views $(1,000)$	$318,975$	$986,581$	$1,387$
# of views traces $(1,000)$	$116,512$	$65,450$	$3,551$
# of referrers traces $(1,000)$	$144,829$	$178,365$	817
	Channel D	Channel E	
Genre	TV trailers	music videos	
Months since established	10	16	
# of videos	914	743	
# of views $(1,000)$	$3,229$	$4,066$	
# of views traces $(1,000)$	$1,454$	$3,623$	
# of referrers traces $(1,000)$	463	$1,514$	

```
2011-10-12,JP,B15e_P2QaTc,Ichiro is stubborn,NO_LINK_MOBILE,,3
2011-10-12,CA,B15e_P2QaTc,Ichiro is stubborn,NO_LINK_EMBEDDED,
video.filestube.com,1
2011-10-12,CA,B15e_P2QaTc,Ichiro is stubborn,NO_LINK_VIRAL,,1
```

We crawled five popular channels with premium content from BroadbandTV's [11] network of over three thousand content partners. BroadbandTV Corp. is a major content aggregator on YouTube, who has distributed tens of thousands of videos and attracted billions of annual views. Due to legal issue and the confidential nature of agreements signed between BroadbandTV and its partners, we cannot disclose the name of the five channels. We therefore name them Channel A to Channel E. Channel A and Channel B are among the most popular channels on YouTube, attracting over hundred millions of annual views. Channel C, Channel D, and Channel E are relatively new, and therefore are less popular than the first two. However, they still attracts about 1.4, 3.9, and 3.0 million annual views, respectively. Table 3.1 summarizes the traces corresponding to these channels. These five channels can represent the popular YouTube entertainment channels, and Table 3.1 also gives a brief description for each channel. Therefore, any YouTube partner with potentially large audience can benefit from the results of this study.

Because the datasets are large, to efficiently conduct our measurement study, we created

a database with MySQL, and imported the datasets into the database. We run SQL queries to obtain partial and aggregate datasets for each measurement. The database schema and the SQL queries utilized in our measurement are listed in Appendix A.

In addition, we examine the Insight data from a global view in this work, that is, we aggregate the worldwide traces in our study. Moreover, we have discovered two obvious outlier videos in two channels. Each video has a sudden burst of number of views which is more than ten times of the normal views of the entire channel. Among the two videos, one video won an award on YouTube and attracted significant greater views in the following two days; the reason of burst of the other video is unknown. To achieve generality of our measurement, we removed the two videos from our study.

3.3 Analysis of YouTube Insight's views Data

In this section, we analyze the data from Insight's views datasets. We examine the statistics of video popularity, and the drop of the tail in views distribution in particular. We propose a simple yet effective method to identify viewing surge, and further characterize the feature. We also study the statistics of visiting behaviour, and investigate user subscription and engagement such as rating, favouriting, and commenting.

3.3.1 Video Popularity

The number of views is one of the most important metrics for YouTube partners, as it is highly correlated with their revenue. The ultimate goal of YouTube partners' everyday operation is to increase views and subsequently generate more revenues. On the other hand, understanding user access pattern is also crucial to YouTube itself, as well as other video sharing websites, as they can enhance their services accordingly.

The views' rank distribution has been examined by most of the video sharing service studies. The distribution is often plotted in log-log scale, and exhibits the "long-tail" property, indicating that there are a few extremely popular videos and a large amount of unpopular videos. The lifetime rank distribution for each channel is shown in Figure 3.2. Our findings are consistent with the literature that the end of the curve of the distribution exhibits clear drop in log-log scale. The drops are quite clear, probably because most of the videos in these channels are popular, compared with random YouTube videos. In fact, very few videos in these channels have been watched fewer than ten times (Channel C, Channel

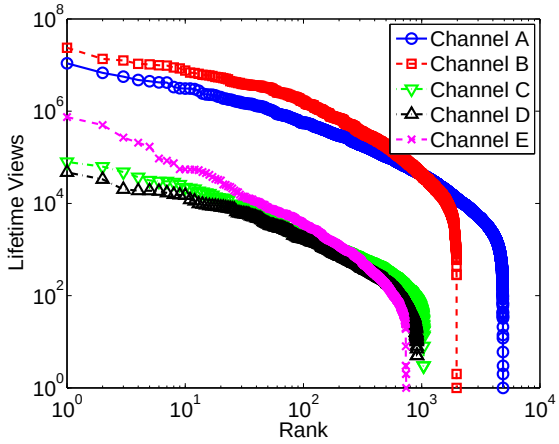


Figure 3.2: Rank distribution of lifetime views

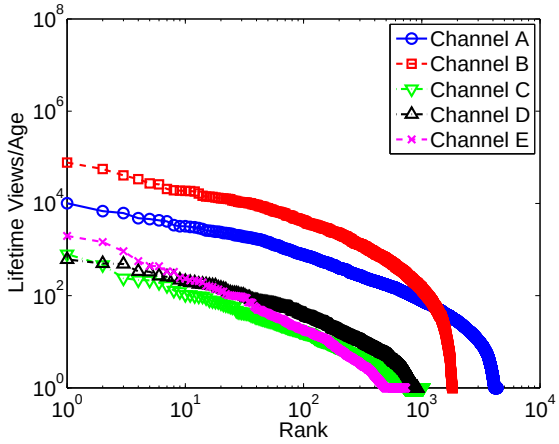


Figure 3.3: Rank distribution of lifetime views per day

D, and Channel E), hundred times (Channel A), and thousand times (Channel B). This is because these channels contain premium high-quality content.

To study the impact of video age, we further divide the number of lifetime views by the video age to get the average views per day. The distribution is shown in Figure 3.3. Compared with Figure 3.2, we observe that the tails of Channel A and Channel B are smoother and less obvious, tails of other channels almost disappear.

To further understand the implication of the drop of the tail, we examined the distribution of views on monthly and daily basis as well. In particular, we extracted 3 consecutive monthly data and 7 consecutive daily data for each channel, and plotted them in Figure 3.4. Even within a small duration (i.e., one day), the drop is clear for Channel B, and is visible for Channel A. On the other hand, there is no drop for the other three channels over a short period, but they appear over a longer period. Moreover, the drop of the monthly distributions for all the channels are less obvious than the all-time distributions. Therefore, the drop appears clearer for longer period. In addition, the degree of decays of the tails also implies the popularity of the channels.

This observation can explain the formation of the drop of the tail for YouTube videos, which is different from the traditional on-demand videos. Even the daily requests follow Zipf's distribution, unpopular videos have more opportunities to be watched than the Zipf's law predicts over longer period, thanks to the social relationship among the videos. In traditional non-social contents distribution, "rich gets richer and poor gets nothing", while

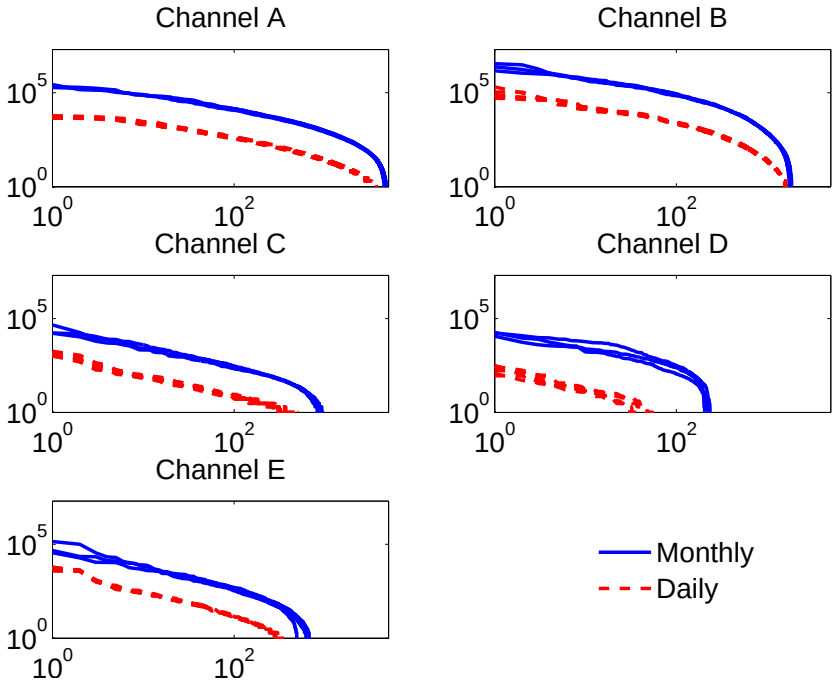


Figure 3.4: Rank distribution of daily and monthly views

in social media, this observation implies that "rich gets richer yet poor still has chance". That means, unpopular videos have potential to be watched and even possibly become popular through YouTube videos' social relationship such as related videos. Therefore, YouTube partners are encouraged to generate more videos that cover a broader range of interest, and a smart user engagement strategy on current unpopular videos may increase their popularity as well.

3.3.2 Viewing Surge

There are a number of research works trying to model the growth pattern of the video views [55]. While we acknowledge that this is an important issue in research area, and the result would be useful for resource provisioning, an accurate, valid, generalized model is however hard to obtain, because there are too many unpredictable factors. For example, YouTube partner can conduct SEO (search engine optimization) on videos by optimizing tags and thumbnails, and this optimization can greatly increase the popularity of the videos; also, if a popular account from a social media website embedded or share a video, the video will

Table 3.2: Pseudo-code of viewing surge detection

Input:
time series of a video V,
surge factor α, threshold β.

Output:
number of surges *surge_no*,
a series of flag indicating surge *surge_location*.

Method:

```
initialize surge_location as all zero;
for each day i
  if first day
    if V(i) ≥ β
      surge_location(i) := 1;
    endif;
    continue;
  endif
  calculate average views of previous min(7, i−1) as avg;
  if V(i) ≥ α · avg and V(i) ≥ β
    surge_location(i) := 1;
  endif
endfor
for i = 2 : end
  if surge_location(i) = 1 and surge_location(i−1) ≠ 0
    surge_location(i) := 2;
  endif
endfor
surge_no := number of (surge_location = 1);
```

gain a great amount of views. These actions can cause a viewing *surge*, and are infeasible to be captured by a generalized model.

Furthermore, for YouTube partners, growth pattern is not important to them, while a viewing surge is crucial to them, as they can gain knowledge about what causes a surge and thus try to create more viewing surges to increase revenues. We therefore provide mechanism to identify viewing surge and further characterize it. Specifically, we utilize a simple yet effective time series analysis to detect the view surge for each single video.

To do that, we first extract a series of views for each video, and examine the number of views along time. Simply said, we compare the views with the previous views to determine if it is a surge. Since the number of views is quite fluctuant, we calculate the average of the

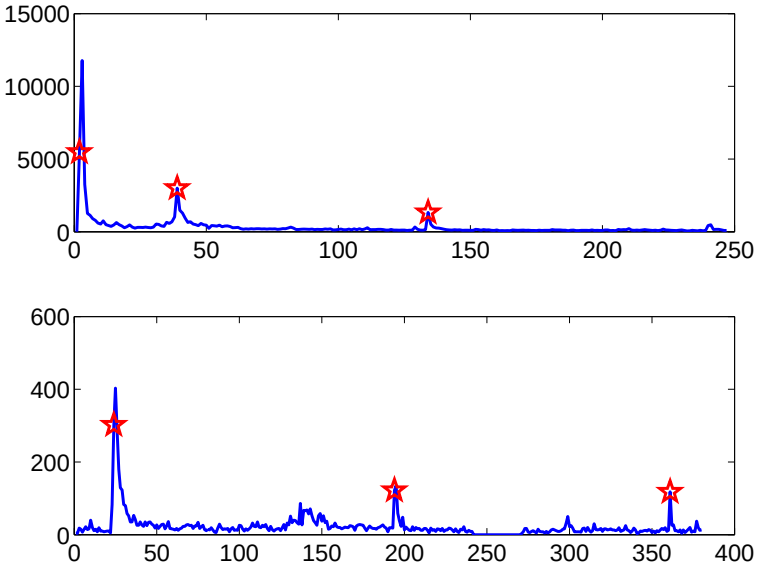
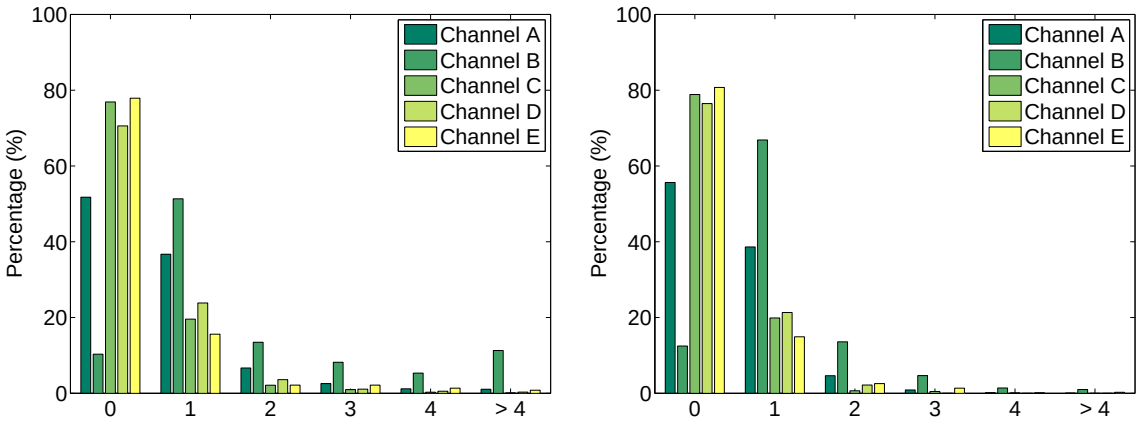


Figure 3.5: Two examples of detecting viewing surges

previous views. In particular, for the views after the first week, if it is greater than α times of the average of the previous 7 days and greater than a threshold β, we define this as a surge; for the views within the first week excluding the first day, we calculate the average of the views before that day; for the views on the first day, we only compare it with β. If consecutive days are surges, we only select the first day as the surge. The pseudo-code of the algorithm detecting view surge for one video is shown in Table 3.2. Two examples of viewing surge detection results are also shown in Figure 3.5.

We have two parameters in the algorithm. One is the surge factor α, which determines the degree of increased views that is considered as a surge. The greater value indicates the more strict becoming a surge. In our algorithm, we test α being 2 and 4. The other parameter is the threshold β, which determines the least number of views that can be consider as a surge. For example, even if the number of views is 20 and the previous average is 5, we do not consider it as a surge, because this small number does not affect the revenue from the YouTube partner point of views. In general, the granularity of monetizing video is thousand views, and thus we set β as one tenth of that, i.e., 100, for Channel C, Channel D, and Channel E; because Channel A and Channel B are extremely popular, we set β as 1000 for these two channels.

We apply the algorithm to each video, and summarize the result for each channel. We

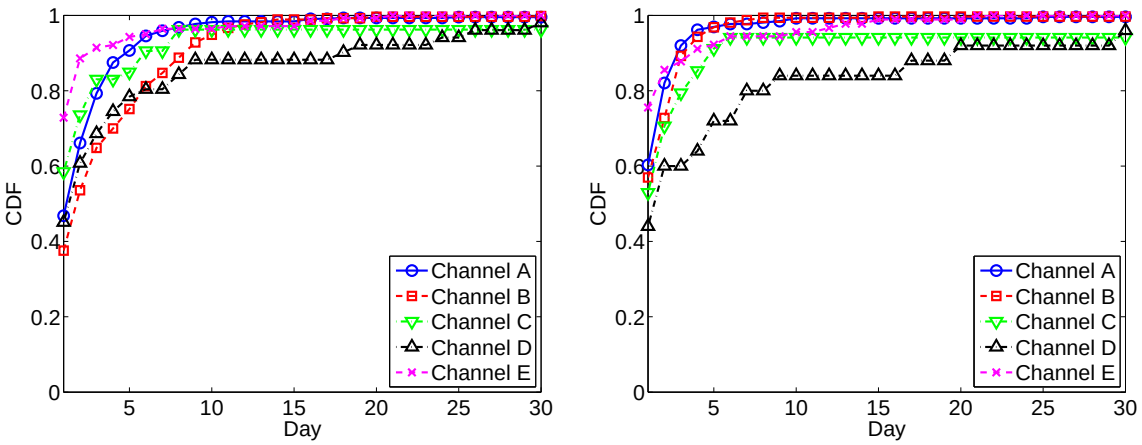


(a) $\alpha = 2$ (b) $\alpha = 4$

Figure 3.6: Breakdown of the number of surges

first look at the distribution of the number of surges, as shown in Figure 3.6 for α being 2 and 4. Except for Channel B, over half of the videos for other channels have no surge detected, probably because the number of views does not reach to the threshold. For Channel B, over 60% of videos have at least one surge, further indicating the popularity of the channel. There are no big difference between different surge factor α for the other four channels, while for Channel B, there are 10% of videos have no less than 5 surges when $\alpha = 2$, yet the number reduces when $\alpha = 4$, implying that many videos in Channel B have several middle level of viewing surges. Moreover, there are 15% of videos in Channel B have two surges. To explain that, the first surge probably occurs soon after the video is uploaded, and since videos in this channel are movie trailers, the second surge probably occurs around the time when the movie is officially released (recall that Channel B videos are movie trailers).

We further examine the location of surges in terms of the date, as shown in Figure 3.7 for α being 2 and 4. We observe that over 40% of surges occur on the first day for all the channels. For Channel A, Channel C, Channel E, almost all the surges occur within the first week. For Channel D, the surges occur later than the other channels. For Channel B, we observe notable difference between different surge factor α. To explain that, the number of views keep increase in the first few days, but not in a great pace that it cannot be detected as a surge for larger α. In summary, the viewing surges mostly occur soon after the video is uploaded, suggesting YouTube partners to utilize this crucial duration to promote their videos, for example, by sharing on social media websites.



(a) $\alpha = 2$ (b) $\alpha = 4$

Figure 3.7: CDF of the location of surges

3.3.3 Daily Pattern

To make smart marketing strategy, viewing pattern is important information to YouTube partners. They can gain knowledge about when the users mostly watch the videos, and therefore upload and promote videos accordingly. Unfortunately, the granularity of the Insight data is day, and thus we are not able to extract the diurnal pattern. We instead extract the daily pattern for each channel.

Figure 3.8 shows the daily pattern. It is clear that Channel A, Channel B, and Channel C have more views during weekends than during weekdays, while Channel D is opposite, and Channel E has evenly distributed views during the whole week. The difference is caused by the variety of video genres. One possible explanation is that videos of Channel D are TV show trailers and TV shows are mostly broadcasted during weekdays, which is different from movies that people are likely to watch during weekends. On the other hand, it implies that it is necessary to differentiate various genres of video, and our analysis extends the overall breakdown in the previous works.

3.3.4 Visiting behaviour

We should note that studying the number of views cannot give us the knowledge about the visitors (for example, how many times a video will be watched by the same user). Traffic log can infer such statistics, but the existing studies are confined by time and region

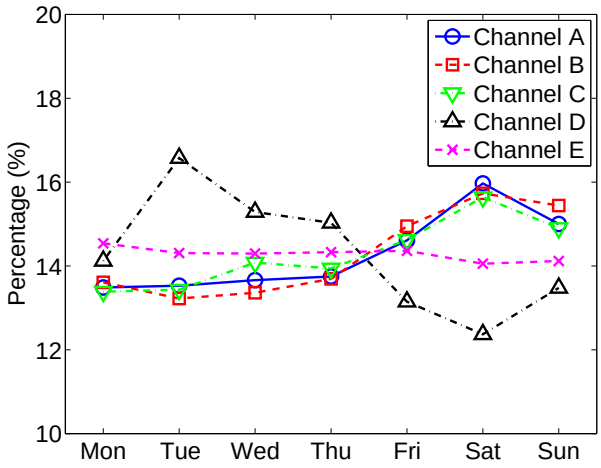


Figure 3.8: Daily viewing pattern

[50]. Fortunately, the views dataset provides these statistics. Not surprisingly, the rank distribution of the number of unique visitors is similar to that of views.

A ratio of the number of views and unique visitors implies on average how many times a video is watched per viewer in one day. We calculated the all-time average ratio for each video and plot the CDF in Figure 3.9. The ratios are on average 1.18, 1.24, 1.47, 1.24, and 1.46 respectively. We can see that Channel B and Channel D are similarly attractive, and both are more attractive than Channel A. On the other hand, Channel C and Channel E are more attractive than the other three.

This is mainly due to the video genre. Recall that videos of Channel A, Channel B, and Channel D are trailers of game, movie, and TV show; users who watch trailers intend to get information about the game, movie, and TV show, and thus they are likely to watch only once when they obtain such information; while videos in Channel C and Channel E are game playing videos and music videos, which are mostly not informative, and if the video content is attractive, users are tend to watch more than once. Therefore, besides the number of videos and number of viewers, the attractiveness of video content is crucial to non-informative video channels such as Channel C and Channel E. Moreover, the characteristics implies that by continuing uploading premium videos, Channel C and Channel E have great potential to attract as many views as Channel B.

Besides the multiple views by the same visitor within one day, we further examine the multiple views within a longer time. The views dataset records the statistics of unique

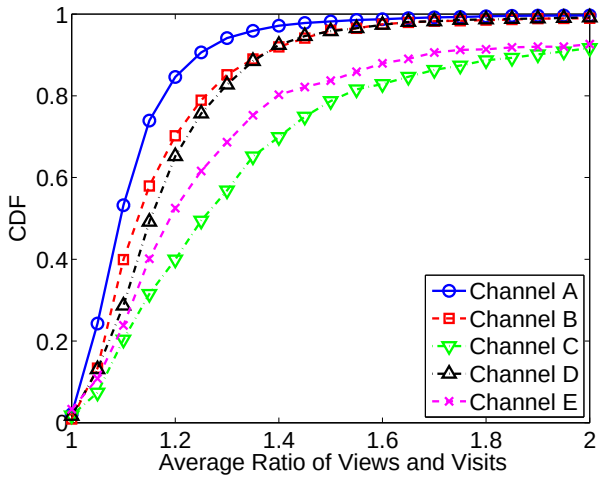


Figure 3.9: CDF of average ratio of views and visits

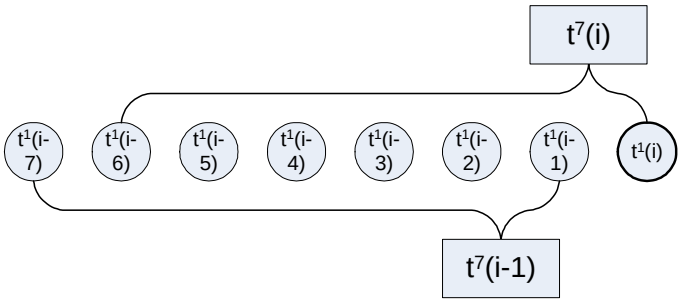
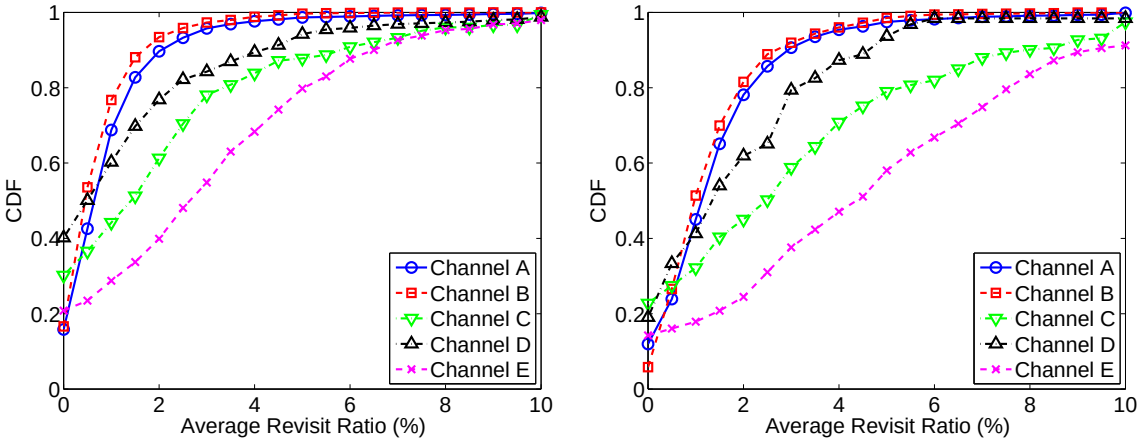


Figure 3.10: Illustration of calculating the number of revisit users within 7 days

visitors in the last 7 days and 30 days. To calculate a ratio of returning visit, *revisit*, we conduct the following processing on the data.

Suppose $t_v(i)$ is defined as the trace of video v at date i, and $t_v^1(i)$, $t_v^7(i)$, and $t_v^{30}(i)$ are video v's statistics of unique visitors on day i, unique visitors in the last 7 days including day i, unique visitors in the last 30 days including day i, respectively. Figure 3.10 shows an illustration of calculating the number of revisit users within 7 days.

Specifically, the number of unique users in the previous 6 days, excluding the current day, is calculated as $t_v^7(i-1)-t_v^1(i-7)$, and thus the number of unique users that have not visited in the previous 7 days, including the current day, is calculated as $t_v^7(i)-(t_v^7(i-1)-t_v^1(i-7))$. We let $rv_v^7(i)$ and $rv_v^{30}(i)$ be the number of revisit user at day i within 7 days and 30 days,



(a) 7 days (b) 30 days

Figure 3.11: CDF of videos' average revisit ratio

respectively. Then we have

$$rv_v^7(i) = t_v^1(i) - (t_v^7(i) - t_v^7(i-1) + t_v^1(i-7)), \tag{3.1}$$

and

$$rv_v^{30}(i) = t_v^1(i) - (t_v^{30}(i) - t_v^{30}(i-1) + t_v^1(i-30)). \tag{3.2}$$

We apply the two equations to each videos, and calculate the *average revisit ratio*. Note that, we have removed those videos that are under 7 days old for revisit ratio of 7 days and those videos that are under 30 days old for revisit ratio of 30 days. Figure 3.11 shows the CDFs of average revisit ratio of 7 and 30 days, respectively. The x-axis represent the percentage of users that will come back to watch the video again within 7 and 30 days. From the figures we can see that there are much more revisit for Channel C and Channel E than the other three trailer channels, which is consistent with the average ratio of views and visit (refer to Figure 3.9).

Since all the partner's videos in the same channel are eligible for monetization, it is also important for the content partner to attract users to come back and watch other videos. Therefore, we further calculate this average revisit ratio for each channel. Note that the Insight data also include the trace for channel, as each trace records the number of various statistics for videos of a channel in a specific region, on a specific day. The method of calculating average revisit ratio for channel is the same. Figure 3.12 shows the average

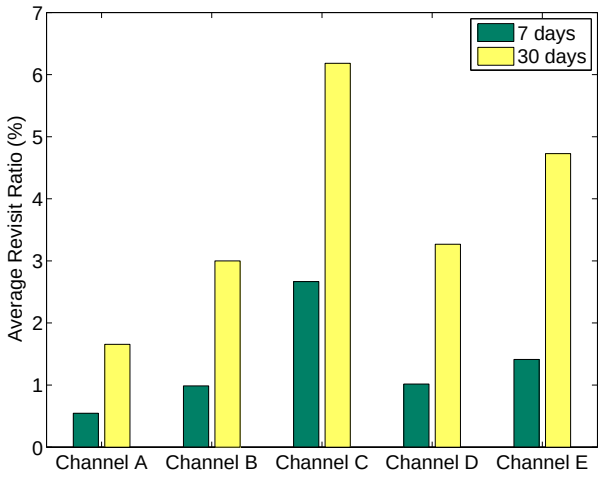


Figure 3.12: CDF of channels' average revisit ratio

revisit ratio of each channel, which is also consistent with videos' average revisit ratio that Channel C and Channel E have higher revisit ratio.

How do the users revisit a channel? A user may be interested in a particular video that s/he has watched before, and thus s/he can search or browse from the watching history to find that video. A better and easier way is to subscribe the channel. Next we will investigate the impact of subscriptions.

3.3.5 Subscription

Users can subscribe to a YouTube channel, so that when they log on to YouTube website, videos from the subscriptions are listed on the frontpage. Subscribers can be considered as the loyal audience of the channel, and thus are crucial to YouTube partners. It is known that users are likely to watch other videos following the current one [21, 24]. As there is a "more from subscriptions" section on the YouTube video page while playing the video from the subscription list, more videos from the same channel have great chance to be watched. This suggests that YouTube partners should upload more videos of similar interest, so that it will attract users to browse more videos of the channel.

Currently, these five channels have 206 thousand, 750 thousand, 23 thousand, 7 thousand, and 12 thousand subscribers, respectively. We calculate the CC (correlation coefficient) between the number of total views and the number of subscribers, and the result is as high

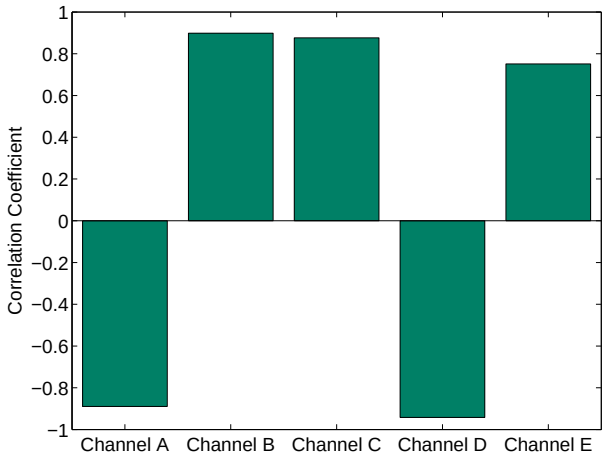


Figure 3.13: Correlation coefficient between subscribers and daily views

as 0.998, which clearly indicates that the popularity of the channels is highly correlated with the number of subscribers.

We further examine the impact of subscribers on each channel along the time. Specifically, we calculate the CCs between the number of subscribers (the sum of the number of subscriptions minus the sum of the number of unsubscriptions) on each day and the average daily views on and after that day, and the results are shown in Figure 3.13. Channel B, Channel C, and Channel E display very high positive correlation, indicating that subscribers of these three channels have great impact on the video views. Interestingly, Channel A and Channel D display very high negative correlation.

To better understand this negative correlation, we plot the number of daily views and subscribers along time for the five channels in Figure 3.14. From the figure we can see that, the number of subscribers of all the five channels are increasing. Daily views of Channel B, Channel C, and Channel E are increasing, while Channel A and Channel D are decreasing from some point. This explains the negative value of CC for Channel A and Channel D.

Regarding Channel D, there had been an event (cannot be disclosed), and this lead to a great decrease of the views, refer to Figure 3.14. This also confirms that generalized model trying to capture the growth pattern of video views are difficult to be accurate, as there are various unexpected events affecting the video views. Nevertheless, it indicates that subscribers for Channel A and Channel D are not so important as the other three channels, as they are not contributing correlated views. It mainly because users reach to the videos

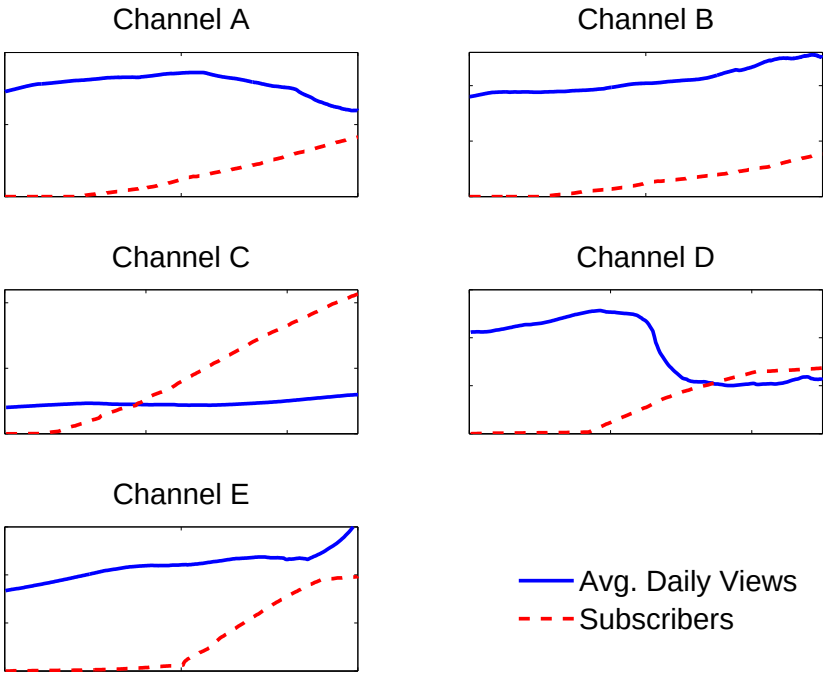


Figure 3.14: Daily views and subscribers along time

of Channel A and Channel D from other means. In addition, this also indicates that various channels have different characteristics.

3.3.6 User Engagement

Users can rate (i.e., like and dislike), favourite, and comment on YouTube videos. Such user activities may make a video more popular. For example, trending videos with many rates, favourites, and comments may be listed in the "most liked", "top favourite", "most commented", and other featured video lists, leading to more popularity. Table 3.3 shows the basic statistic of rates, favourites, and comments of each channel.

To understand the user behaviour on engagement, we calculate a value of views per engagement. The results are shown in Figure 3.15. We can see that Channel A, Channel B, and Channel C are similar: they have similar views/rates and views/comments ratio, and views/favourites ratio is nearly twice of the value of the other two ratios. This implies that videos of these three channels have comparable number of rates and comments, and the number of favourites is smaller. Channel D and Channel E exhibit different characteristics

Table 3.3: Statistics of Rates, Favourites, and Comments

channel	engagment	**max**	**median**	**mean**	**deviation**
Channel A	rate	20,958	50	148	568
	favourite	22,043	20	106	546
	comment	22,483	45	172	799
Channel B	rate	44,524	225	853	2,358
	favourite	36,712	95	585	1,957
	comment	59,433	164	1,067	3,618
Channel C	rate	621	2	7	24
	favourite	105	1	2	6
	comment	6,580	1	27	287
Channel D	rate	82	1	3	7
	favourite	58	0	1	4
	comment	26	0	1	2
Channel E	rate	4,754	5	37	248
	favourite	5,741	2	37	284
	comment	1,273	1	11	72

from the other three channels: the main difference is that the views/comments ratio is much larger, indicating that comments on videos of the two channels are relatively rare. In addition, Channel E has the smallest views/raties and views/favourites ratio, indicating that users are more willing to engage with the videos of Channel E.

A recent study on Facebook shown that the average clicks of a post per comment is four times greater than average clicks per "like" [112]. However, as shown in Figure 3.15, YouTube displays different characteristics. As discussed, the first three channels have similar number of views per rate and comment; views per comment is over 2 times and 5 greater than views per rate for Channel D and Channel E, respectively. Moreover, the number of views per rate and comment (over hundreds) is much larger than that in Facebook (3.1 and 14.7). The large values indicate that rating and commenting are rarer on YouTube than on Facebook. This is perhaps due to the main difference between video sharing services and social networking services: YouTube videos are watched by worldwide users, while Facebook posts are mostly clicked by friends or fans. It is logical that because of the more social nature of Facebook, user engagement on this platform is more than YouTube.

We also calculate the CC between views and rates/favourites/comments, shown in Figure 3.16. We can see from the figure, CCs between views and all three engagements for

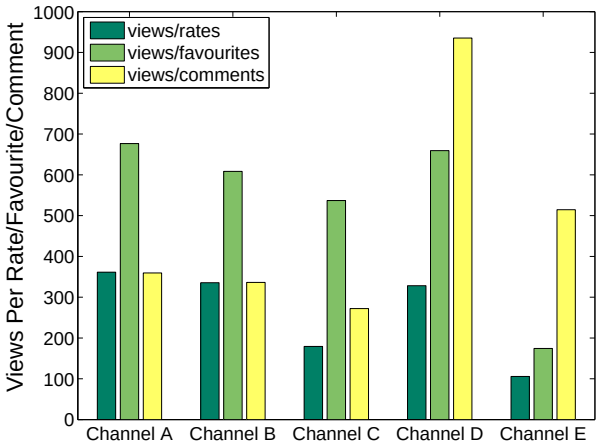


Figure 3.15: Views per rate/favourite/comment

Channel A and Channel B are very high; CCs between views and rates/favourites for Channel E are near 1, yet views and comments shows some degree of correlation. CCs for Channel C and Channel D are not as high as other three channels, but still show some degree of positive correlation, except views and comments for Channel C. Nevertheless, the relatively high correlation coefficient between video views and user engagements implies that a video has a better chance to be watched if it is engaged more. From this, YouTube partners are suggested to promote the videos by leading user engagement activities such as discussion and poll, making users involved.

3.4 Analysis of YouTube Insight's referrers Data

YouTube users have various means to reach YouTube videos. The last webpages where the viewers come from is called *referral sources*. Understanding referrals is essential for YouTube partners to adapt their user engagement strategy. The referrers dataset provides this valuable information. We can classify the referral sources into four categories:

SUGGESTION. The referral came from YouTube's related video links;

VIDEO SEARCH. The referral came from YouTube or Google search results;

YOUTUBE SURFING. The referral came from any YouTube pages except related video links and search results, including annotation links, YouTube channel pages, subscriber

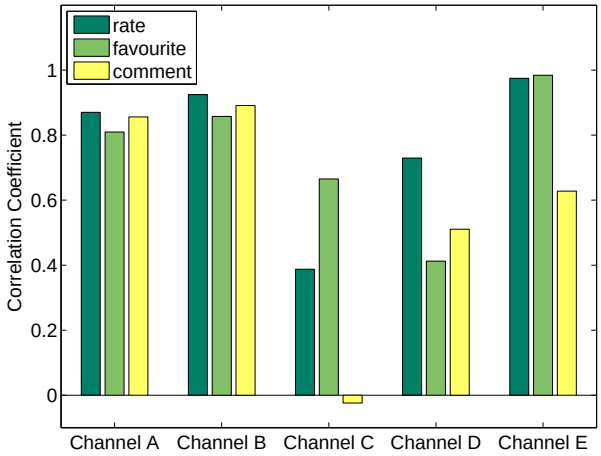


Figure 3.16: Correlation coefficient (CC) between views and rates/favourites/comments

links, paid and unpaid YouTube promotion, and other pages on YouTube;

SOCIAL REFERRAL. The referral source was a link on an external web page, or the video was embedded on an external web page;

NON-SOCIAL DIRECT. YouTube analytics did not identify a referral source, indicating that the viewer navigated directly to the video, e.g., by copying and pasting the URL.

3.4.1 Referral Sources

Figure 3.17 shows the breakdown of the above five categories for each channel. It is clear that the breakdown percentages are channel-dependent. For example, one-third of the users reach Channel A videos from suggested videos, and one-third reach from search results; Channel B and Channel D is similar to Channel A; very few users reach Channel C videos from external sources; half of the users reach Channel E videos from search results.

This observation confirms and extends the findings in the previous works [41, 127] that search results and related videos are the top sources of views. In addition, the large portion of referrals from SUGGESTION and YOUTUBE SURFING confirms our hypothesis of subscribers will watch more videos as discussed in Section 3.3.

This breakdown provides YouTube partners helpful knowledge about the traffic source and audience behaviour of their channels.

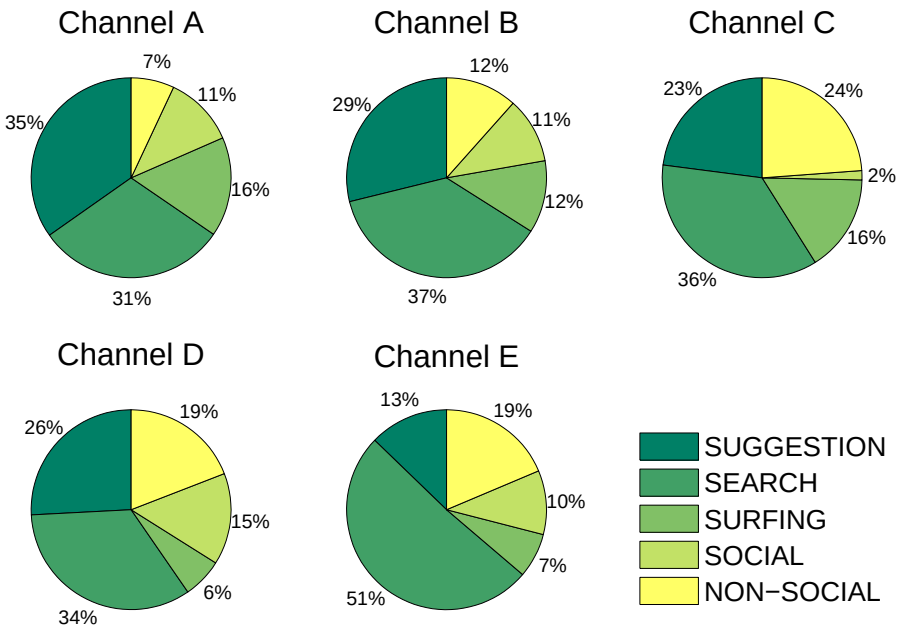


Figure 3.17: Breakdown of the referral source

- With respect to YOUTUBE SURFING, user watching behaviour mainly depends on YouTube service, such as the playlists and trending videos. Moreover, related videos (SUGGESTION) are also affected by the ranking algorithm of YouTube. Therefore, YouTube partners are suggested to abide by guidelines specified in YouTube's Creator Playbook [122], which impacts the ranking of their videos.
- From SOCIAL REFERRAL, YouTube partners can gain knowledge about which external websites contribute more views. Thus the partners can further establish relationships with those websites, or create a new relationship with the websites that are not currently bringing in much external traffic.
- From VIDEO SEARCH, correlation between search keywords and video popularity is worth studying. The YouTube partners can improve the search rank of their videos by tagging those videos with popular yet relevant keywords. This interesting topic is out of the scope of this thesis, and we will investigate it in the future studies.
- No further information is currently available for NON-SOCIAL DIRECT, and thus we cannot investigate this category further.

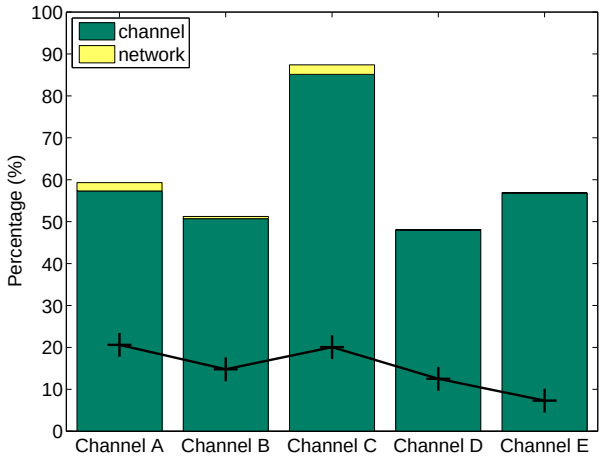


Figure 3.18: Percentage of referral related videos from the same channel/network

In the following, we specifically investigate impact of SUGGESTION and SOCIAL REFERRAL on video views.

3.4.2 YouTube Suggestion

Since SUGGESTION is one of the most important referrals, we examine how related videos contribute video views. In the referrers dataset, the video IDs of the referral related videos are recorded. Therefore, we calculate the percentage of referral related videos that are from the same channel, that is, the referral related videos are the YouTube partner's own videos. It is known that a YouTube partner can have multiple channels and organizes these channels as a network. In short, YouTube partners' channels are correlated. Therefore, we further calculate the percentage of referral related videos that are not from the same channel but from the same content network, that is, from the other four channels in our data.

Figure 3.18 shows the results. We can see that over half of the referral related videos are from the same channel, implying the importance of the related videos for the YouTube partner. We also find that the percentage for Channel C is much higher than the other four channels, reaching over 85%, suggesting that related videos are more important for this channel. There are also small percentage of referral related videos from different channels that are within the same content network, confirming that other channels in the same content network do have chance to contribute video views.

We further calculate the ratio between referral related videos that are from the same

Table 3.4: Summary of Top External Websites Referrers

	Channel A	Channel B	Channel C
1st	9.0% downloading site	16.2% Facebook	31.9% gaming wiki
2nd	4.4% Facebook	2.2% n/a	7.6% Facebook
3rd	2.6% forum	1.5% downloading site	5.3% gaming blog
4th	1.7% gaming site	1.2% n/a	5.1% gaming site
5th	1.5% gaming site	0.9% downloading site	3.7% Internet video site
	Channel D	Channel E	
1st	41.2% Reddit	62.4% Facebook	
2nd	9.9% Facebook	2.4% music streaming	
3rd	4.7% Twitter	2.0% music blog	
4th	2.0% blog	2.0% Twitter	
5th	1.7% entertainment site	1.6% music blog	

content network and the total views, as the line shown in Figure 3.18. The ratios are between 0.1 and 0.2, implying the various importance of video suggestion for different channels. For example, the ratio for Channel E is relatively low, because the most important view source is search result, referred to Figure 3.17. Nevertheless, video suggestion is an important view source, and thus YouTube partners are encouraged to upload more relevant videos.

On the other hand, however, videos in the related list are not necessarily from the same channel. This ranking mechanism of the related videos depends on YouTube service. Although the detail of the algorithm is not public, YouTube provides some guideline of how it organizes related videos, and YouTube adjusts the ranking algorithm periodically to augment better services. Therefore, the YouTube partners are suggested to check the latest guideline, understand and adapt to it. There are also some methods to inter-connect their videos so that the videos have better chance to be listed in the suggestion, for example, inserting annotation, creating play-list. Moreover, YouTube partners are encouraged to deploy videos covering a broad range of interests.

3.4.3 External Website Referral

Although SOCIAL REFERRAL is not the top view source, it does not indicate that external website referral can be ignored. Similar to the impact of subscriptions, there is a great chance that users will watch more videos from the related video list after finish watching one. Therefore, SOCIAL REFERRAL can be considered as an introductive referral.

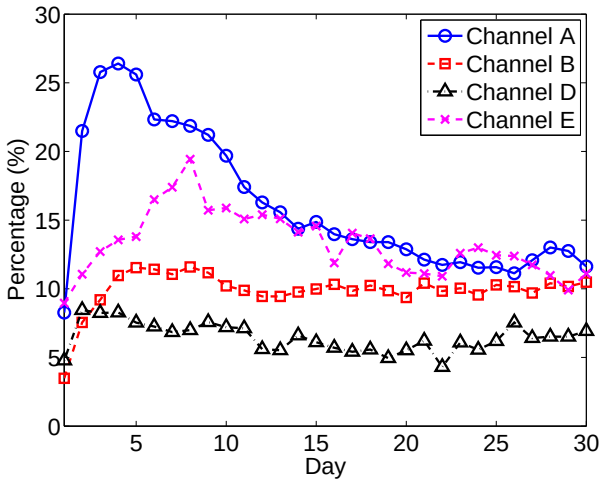


Figure 3.19: Average percentage of views from SOCIAL REFERRAL in the first 30 days

Table 3.4 lists the top-five external website sources for each channel. We do not disclose the specific names of the websites except such notable social networking service as Facebook, Twitter, and Reddit [86], and simply use general descriptions. From the table, we observe the channel-dependency characteristic again. No external website dominates the external referrals for Channel A, and the small percentage indicates that there is a great number of sources. In Channel E, over 60% of the referrals are from Facebook, over 20 times greater than the second one. Facebook also dominates in Channel B, yet the percentage is not as high as in Channel E. Facebook is the second in both Channel C and Channel D, while the first ones have high percentage. In summary, this result gives YouTube partners a clearer strategy for promoting their videos, e.g., by establishing relationship with those websites.

To understand the impact of external website referrals at the different stages of a video, we specifically looked at the percentage of views from SOCIAL REFERRAL on each day for the first 30 days after the video was uploaded, and calculate the average. The results for each channel are shown in Figure 3.19. Note that, since only 2% of views came from external sources for Channel C (refer to Figure 3.17), we do not include this channel.

All the four channel have one same feature: they have notable increasing trend at the first few days, and then start to decrease or keep constant. Channel A, Channel B, and Channel D reach to the peak at the fourth, fifth, and second day, while Channel E reaches to the peak later, at the eighth day. To explain that, the first three channels contain trailer videos of

various genres, so it is very likely that users are expecting such new games, movies, and TV shows, and thus the videos are soon extensively spreading in the external websites; while Channel E contains music videos from independent artists (different from major commercial record labels), and thus the videos need more time to be noticed.

In addition, the peak values of Channel A and Channel E are greater than that of Channel B and Channel D, and is much higher than the overall percentage (recall that in Figure 3.17, the percentage of SOCIAL REFERRAL views for the four channels are 11%, 11%, 15%, and 10%, respectively), implying that external sources are important to these two channel in the early stage, and this suggests YouTube partners to promote their videos to create viewing surges soon after uploading. Moreover, the percentage of Channel B is almost constant and is similar to the overall percentage (11%) after reaching the peak, indicating that although not driving as much percentage of traffic as Channel A and Channel E do, external sources remain impacting the video views for a long time. On the other hand, the percentage of Channel D is much lower than the overall percentage (15%), mainly because some very popular videos have more SOCIAL REFERRAL views, as the deviation of the result is quite large (between 10% and 15%).

3.5 Conclusion

YouTube partners have changed YouTube content distribution landscape by uploading and monetizing premium videos. YouTube provides Insight analytics to partners, and the Insight analytics show simple scalars and charts related to videos' traffic and audience. However, further analysis is needed so that YouTube partners can benefit from adapting content deployment and user engagement strategies. In this chapter, from up to three years of Insight data of five YouTube partners' channels, we have revealed a number of unique features of YouTube partners, such as further understanding of the drop of the tail in views distribution, statistics of viewing surges, daily pattern, characteristics of visiting behaviour, as well as the impact of user subscription and engagement on video views, extending the exist knowledge about video sharing service in the literature. We have also studied the breakdown of referral sources, and specifically investigated the impact of video suggestion and external website referral on video views. This study is of great value to YouTube partners for attracting more views and subsequently generating more revenue.

Chapter 4

Video Spreading in Social Networks

The modern social networking services have drastically changed the information distribution landscape and people's daily life. With the development in broadband accesses, video has become one of the most important types of object spreading among social network users. Yet the sheer data volume and the long access durations of video objects present more significant challenges than other types of objects, not only to the social networking service management, but also to the network traffic engineering.

In this chapter, we take an important step towards understanding the characteristics of video spreading in the social networks. Our study is based on real data traces from Renren [87], the largest online social networking service in China. We present our data collection methodology in Section 4.2. We examine the user behaviour from diverse aspects and evaluate different users' activities in Section 4.3. In Section 4.4, we also examine the temporal distribution during spreading as well as the typical propagation structures, revealing more details beyond stationary coverage. In Section 4.5, we further extend the conventional epidemic models to accommodate the diversity of the spreading, and our model effectively captures the process of video spreading in the social networks, serving as a valuable tool for such applications as workload synthesis, traffic prediction, and resource provisioning.

4.1 Introduction

In the past decade, particularly since the emergence of Facebook and Twitter, the information distribution landscape and even people's daily life have drastically changed. These online social network services directly connect people through cascaded relations, and information thus spread much faster and more extensively than through conventional web portals or newsgroup services, not to mention the cumbersome emails [107]. As an example, Twitter first exposed the breaking news of Bin Laden's death, 20 minutes before officially confirmed [90]; it also reported Tiger Woods' car crash 30 minutes before CNN [93], inverting the conventional 2.5-hour delay of online blogging after mainstream news report [65].

There have been pioneer studies on information propagation over generic networks [36, 107] and, more recently, over social networks [12, 14, 98]. Yet their focuses have been largely on the conventional text or image objects and on their stationary coverage among users. With the development in broadband accesses and data compression, video has become an important type of objects spreading over social networks, and today's video sites have also enabled social feeds that automatically post video links to user's social networking site personal pages. Video objects, as richer media, however possess quite different characteristics. From data volume perspective, video objects are generally of much larger size than other types of objects; hence, most videos are fed from external hosting sites, like YouTube, and then spread as URL links (together with titles and/or thumbnails). Video sharing thus involves not only internal information propagation in the social network, but also external data accesses. As reported, an auto-shared tweet results in 6 new `youtube.com` sessions on average, and over 500 tweets per minute containing a YouTube link [125].

From social perspective, text diaries and photos often possess personal information, while videos are generally more "public". Together with the shorter links, videos often spread more broadly than texts and images. This new video spreading trend has brought up numerous well-known Internet memes [101] and such celebrities as Justin Bieber [124]. Yet the sheer and ever increasing data volume, the broader coverage, and the longer access durations of video objects also present significant challenges than other types of objects, not only to the social networking service management, but also to the network traffic engineering and to the resource provisioning of external video sites.

In this chapter, we take an important step towards understanding the characteristics of video spreading in social networking websites. Our study is based on one week of 12.8 million

video sharing and 115 million viewing event traces from Renren, the largest Facebook-like online social networking service in China. We examine the user behaviour from diverse aspects, and identify different types of users and evaluate their activities. We also examine the temporal distribution during spreading as well as the typical spreading structures, and reveal more details beyond stationary coverage. We further introduce an SI^2RP Model which extends the conventional epidemic models to accommodate diverse types of users and their probabilistic viewing and sharing behaviour. We validate our model and show that it effectively captures the process of video spreading in social networks. Therefore, the model can serve as a valuable foundation for such applications as workload synthesis, traffic prediction, and resource provision of video servers.

4.2 Data Collection

4.2.1 Background and Methodology

The Renren Network [87] is a Facebook-like social networking service in China. Starting from December 2005 dedicated to campus students, it has dramatically expanded its scale, becoming the largest online social networking service in China. Today it owns 160 million registered users, attracting 31 million monthly user accesses [88].

Collaborating with Renren's engineers, we have extracted the logs on contents shares and video accesses from Renren's server farm. From the sheer volume of logs, we find that the records exhibit noticeable daily patterns, i.e., the characteristics in different days are the same, showing repetitive patterns. In addition, from the measurement in the next section, we find that a shared video will only be active within 5 days; after that, most of the videos will stop spreading no longer be watched. Therefore, we will focus on the measurement results for one week, from 00:00:00 of March 24th to 23:59:59 of 30th, 2011.

Since our data traces are from the entire website, there is no bias in the data like other works that randomly crawl and sample the data. Moreover, as Renren is the largest and most popular online social networking service in China, our study can be extended to other social networking services in the world.

4.2.2 Data Format

For ease of exposition, we first list some key terminologies. When a user posts a video link from an external video sharing website to the social networking website, we refer to the action as *initiate*, and the user as *initiator*. A user can *share* a video after watching it; unless otherwise specified, "share" does not include "initiate", yet "shared contents" includes "initiated contents".

Two types of datasets are obtained for our study.

- In *sharing dataset*, a record was logged when a user clicks the "share" button for a content in her/his Renren portal, including diary, album and video. The record includes the content type, shareUserId, shareId, and time. content type shows the type of shared content, and in this thesis, we only interested in video contents, containing 12.8 million video sharing records, among over 29 million total records. In Renren network, each user is assigned a hashed ID, and shareUserId is the ID of the user who shared the content. When a user shared a content, including initiate, a shareId was created for the share; note that, if a content was shared multiple times, it is corresponding to multiple shareIds. Finally, the record also logged the timestamp when the share occured. An example sharing record is like:
  `create type:2 shareId:5659000000 shareUserId:263000000`
  `2011-03-24 00:00:00.`

- In *viewing dataset*, there are over 115 million records, each of which was logged when a user started watching a video. The record includes the video URL, viewerId, shareId, shareUserId, fromUserId, sourceUserId, and time. viewerId is the ID of the user who watched the video. shareId is the ID of the shared video, and shareUserId is the ID of the user who shared the video and watched by viewerId. It is probable that shareUserId's friend had shared the video before shareUserId further shared it, and thus shareUserId's friend who shared the video is fromUserId. The initiator was logged as sourceUserId. The record also logged the URL of the video and the timestamp when the user watched. An example viewing record is like:
  `http://v.youku.com/v_show/id_XXXXXXXXXXX.html`
  `viewerId:714000000 shareId:5643000000 shareUserId:714000000`
  `fromUserId:241000000 sourceUserId:264000000 Time:2011-03-24 00:00:00.`

4.2.3 Data Pre-processing

There are a series of pre-processing steps to be done for the raw dataset. The first major challenge is that whenever a user shares a content, a unique shareId is created; yet URL is not included in the sharing dataset. As a result, one video may correspond to multiple shareIds, and we need to associate the two datasets to get the URL for each shareId.

The second issue is that, the number of received videos is not readily available. By merging records with the same viewerId from the viewing dataset, we get the friends of each viewer, and by merging records with the same shareUserId, we get the number of videos shared by each user. Finally, for each viewer, we summate the number of videos shared by each friend to obtain the estimated number of received videos.

Finally, it is necessary to associate user's viewing and sharing records, i.e., to extract the records from the viewing dataset and the records with the same shared video by the same user from the sharing dataset. Since URL is not available in sharing dataset, we associate each URL to a number of shareIds from the viewing data. We finally put URL and the user id (viewerId of the viewing record and shareUserId of the sharing record) together, and find the corresponding viewing and sharing record.

Note that, all of the above pre-processing steps are non-trivial, because the amount of the data are too large to be held in memory, and thus we have to process them in a distributed manner.

4.3 User behaviour in Video Spreading

We first investigate the behaviour of individual users on spreading videos in the social networking website. We are particularly interested in the following three key questions:

1. How users initiate video sharing;
2. How users react upon receiving shared videos from friends, i.e., to watch or not;
3. How users react upon finishing watching videos, i.e., to share or not.

A natural hypothesis here is that if a user is active, s/he will initiate many videos, and also watch many and share many. Our findings however show that this hypothesis is not true; in particular, the three behaviours are not necessarily correlated.

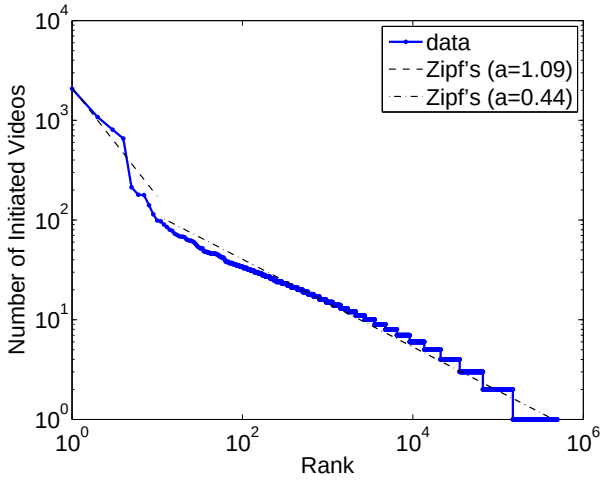


Figure 4.1: Rank distribution of initiated videos

4.3.1 Initiating

We start from examining initiators, each of which triggers the first share of a video. From the dataset, we extract 827 thousand initiating records. While this number is not small, it is only 6.5% out of the 12.8 million sharing records. This indeed reflects the pervasiveness and power of video spreading in social network.

The rank distribution of the initiators (in terms of the number of initiated videos) is plotted in Figure 4.1. Without surprise, it is long-tail and scale-free, suggesting that most users initiate few videos, but a few *active users* have initiated a remarkable number of videos. The most active user indeed has initiated over two thousand videos in one week.

The Zipf's law [128] is usually used to fit long-tail, and the distribution is a straight line in logarithmic scale. However, our data cannot be simply fitted by one Zipf's line: the data after top-10 appear to be a straight line, but the top-10 data clearly differ from the rest. Yet they can be roughly fitted by another Zipf's line. The distinction suggests the existence of two possible types of users with different initiating behaviours.

To better understand this, we calculate the number of users that have directly watched each initiator's video. This number reflects the influence of a user, and estimate the approximate number of friends a user have. We examine the correlation (CC) between the number of initiated videos and the number of friends. Strangely, the CC is extremely low, which is nearly 0 across all initiators. We then generate a series of dataset samples, where each

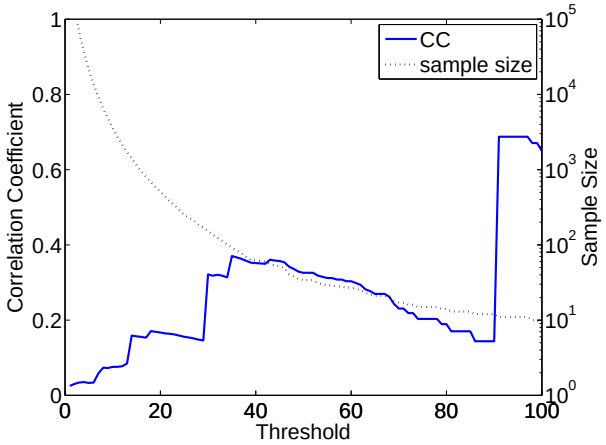


Figure 4.2: Correlation coefficient of the number of initiations and friends

sample eliminates the users that have initiated videos less than a threshold. Thus the larger the threshold is, the smaller sample size is, and the more videos the users have initiated. We compute a series of CCs of each sample, as shown in Figure 4.2; note that we ignore the data with threshold larger than 100 because the sample size, which is also plotted in the figure, is too small (below 10) to be accurate. From the figure we can see the CC is between 0 and 0.4, and suddenly increases to over 0.7 when the threshold reaches about 90.

This result clearly shows that there are two types of initiators: most of the initiators (over 99%) initiate only a few videos and the number has little correlation with their friend population (the CC is 0.22 on average and 0.4 at most); on the other hand, a set of active initiators have much more friends and also initiate a much larger number of videos. The turn at threshold 90 is actually quite sharp (see Figure 4.1), showing clear distinction of the two types of initiators. Since these active initiators serve as hubs that draw much more attention than the general users, they are worth particular attention in system optimization; also, they are valuable to the partners of video sharing websites (refer to Chapter3), as they have great impact on the video popularity.

4.3.2 Receiving and Watching

When a user shares a video, the friends will be notified in the news feed on the webpage. Different from text or images that can be instantly viewed, a shared video will not be really watched until the recipient clicks the link. How users react to the shared videos in the news

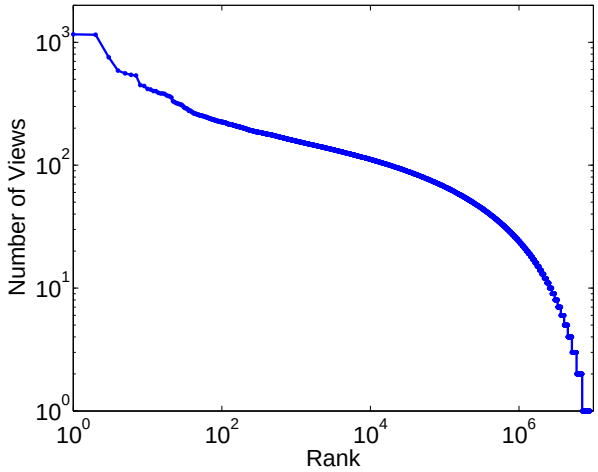


Figure 4.3: Rank distribution of watched videos

feed, i.e., to watch or not, is thus a crucial step to the video spreading.

We first examine the number of videos a user watched. Not surprisingly, the distribution is highly skewed, as shown in Figure 4.3. To understand user behaviour on watching video, besides the number of videos watched, we should also consider the number of videos that the user received from friends, given that the activenesses of user's friends are various. Although the number of shared videos appearing in each user's news feed is not available from the dataset, we can estimate this statistic as discussed in Section 4.2.

We compute the ratio of the number of viewed videos and that of the received videos from friends, defined as the *reception rate* for each user. Since the number of received videos is estimated and the accuracy would be affected if the sample size is small, we removed those users that receive less than 4 videos to obtain a more representative result, and the cumulative distribution function (CDF) of the reception rate is shown as the blue line in Figure 4.4. This curve can be well fitted by the generalized Pareto distribution (GPD). On average, we find that a user watches 16% of videos shared from friends.

Obviously, whether a particular video will be viewed depends on the attractiveness of the video title and/or thumbnail to the particular user and the activeness of the user. The former is out of the scope of this thesis. To this end, we evaluate the correlation between the reception rate and the number of initiated videos. To our surprise, the two characters are nearly non-correlated. Again, we compute the series of CCs against threshold, as we did for that of initiation. The result is shown as the blue solid line in Figure 4.5. For most

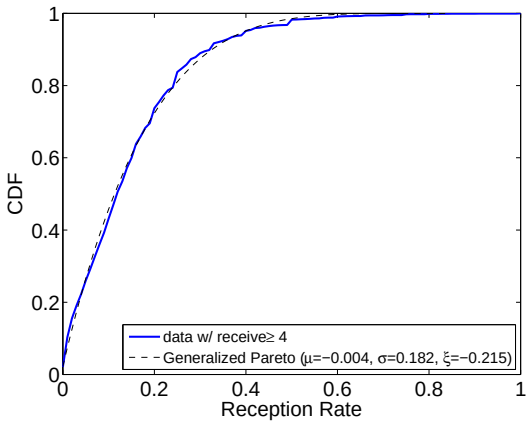


Figure 4.4: CDF of reception rate

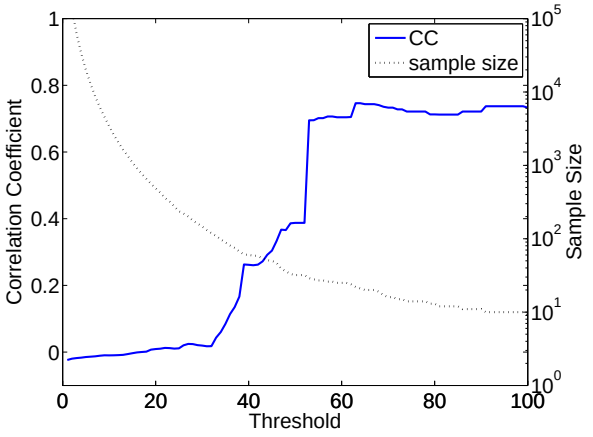


Figure 4.5: Correlation coefficient of reception rate and the number of initiations

of the users with few initiations, the correlation coefficients are near zero, indicating that those user's behaviour of initiating videos has nothing to do with watching videos. Yet for tens of users that have initiated more than 50 videos, the correlation coefficients are quite high (near 0.8). This finding suggests us that activeness is not necessarily a factor that determines user behaviour on initiating and watching videos in the social network websites.

4.3.3 Sharing

We next examine the user behaviour on sharing videos after watching, a key step toward video spreading. The distribution of the number of each user's shares is again long-tail and scale-free, as shown in Figure 4.6, clearly indicating that there are some extremely active users sharing a great number of videos, and most of the users only share a small number of videos. To understand how a user reacts upon finishing watching a video, that is, whether or not to further spread the video, we calculate the ratio of the number of shared videos against the number of watched videos for each user, defined as the *share rate*. In the calculation, we include the users that have not shared any videos but have watched at least two videos. For the cases where the number of shares is greater than the number of views, the share rate is defined as 1.

The CDF of share rate is shown as blue solid line in Figure 4.7. Similar to the reception rate, the share rate can be well fitted by a generalized Pareto distribution. We notice that there are over half of the users do not share any video. We further examine these users by

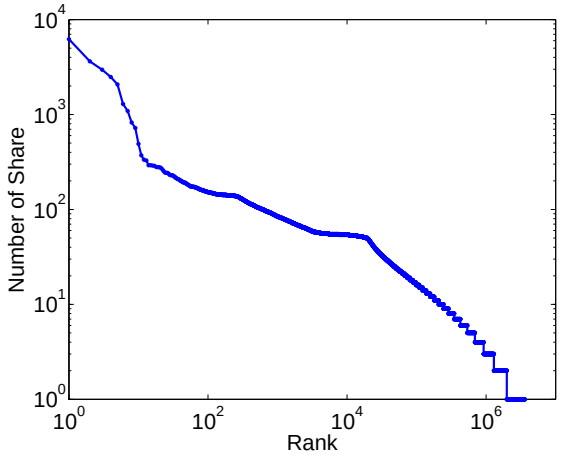


Figure 4.6: Rank distribution of shared videos

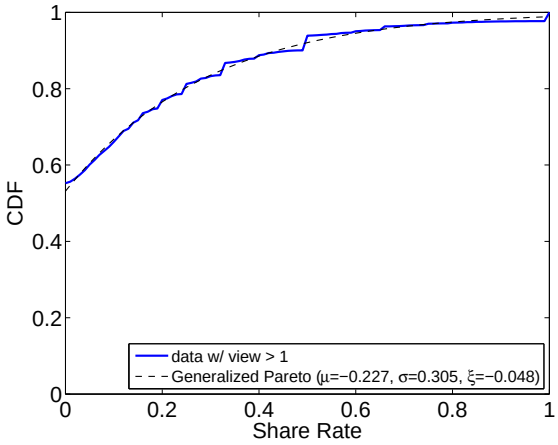


Figure 4.7: CDF of share rate

plotting a CDF of the number of views in Figure 4.8. Among these users, most of them have only watch a few videos, but we do find the most "selfish" users have watched more than one thousand videos without sharing any. Such *free-riders*, like those in peer-to-peer systems, largely hinders the video spreading.

It is well known that a power-law distribution usually results in "$K/(100-K)$" rules, such as 80/20, 90/10, and 99/1, indicating that majority of the effects come from minority of the causes. We thus utilize this method to identify free-riders. Specifically, we only examine the free-riders with views between 1 and 388[1]. As a result, we find a 94.5/5.5 rule, i.e., 94.5% users have watched fewer than 5.5% videos, which is $388 \cdot 5.5\% = 22$. We define users that have watched more than 22 videos without sharing one as free-riders, which are around 320 thousand and count for about 3.5% of the all observed users.

To further investigate whether user behaviour on watching and sharing videos are correlated, we generate a series of samples, each eliminates the users that have shared videos less than a threshold. We compute the CCs between the two rates, shown by the blue solid line in Figure 4.9, and we also show the series of CCs between the number of shares and that of the views, by the red dash line. We observe the CCs of the two rates are above 0.6 for most of the users, showing some degree of correlation; they then drop to 0.2 afterwards, and suddenly drop to near zero for users that have shared near 300 videos. On the other

[1]One outlier free-rider has watched 1151 videos, and all other have watched no more than 388.

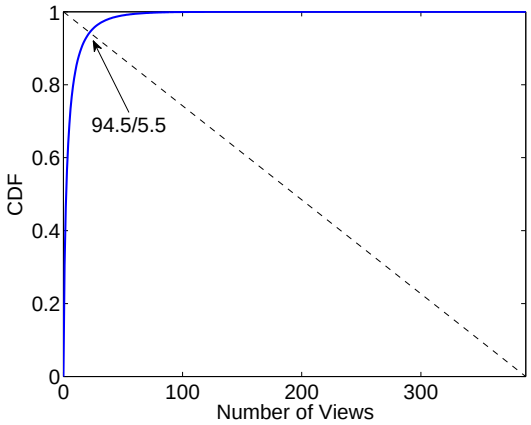


Figure 4.8: CDF of views for free-riders

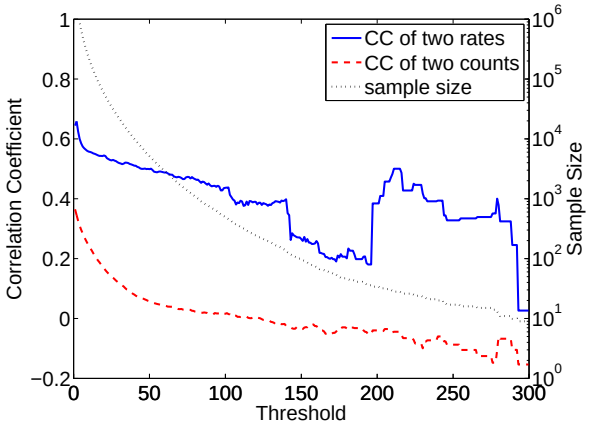


Figure 4.9: Correlation coefficient of share rate and reception rate, share count and view count

hand, the CC of the two counts decreases from 0.4 to -0.2 as the minimum number of shares increases, suggesting that for most users that share several videos, the viewing behaviour and sharing behaviour are loosely correlated, but for other users, the two behaviours are not correlated or even negatively loosely correlated. This also means that, extremely active users that share hundreds of videos in fact do not necessarily watch that many.

4.3.4 Summary

The above observations suggest that the users have diverse activenesses, but they are not necessarily correlated. Though in this case it is difficult to find a universal model for characterizing the behaviour of all users, we can roughly distinguish three types of users.

First, a small number of users initiates a lot of videos, and also have many friends, being hub-like. These *spreaders* (SU) are critical to the start of video spreading. Through examining Renren data, we find that spreader users are often non-personal accounts specifically in collecting and spreading interesting, funny, attracting contents, including videos; it is also possible that spreaders are bots, spreading videos in a spam manner.

Second, *free-riders* (FU) that watch many videos without sharing any, which noticeably hinders the video spreading. As estimated, there are approximate 3.5% free-riders.

We call the rest *ordinary users* (OU), as they sometimes initiate a few videos, watch some shared videos, and occasionally share some videos they watched. The three behaviours can be different that some users may be only active in watching videos while some may be

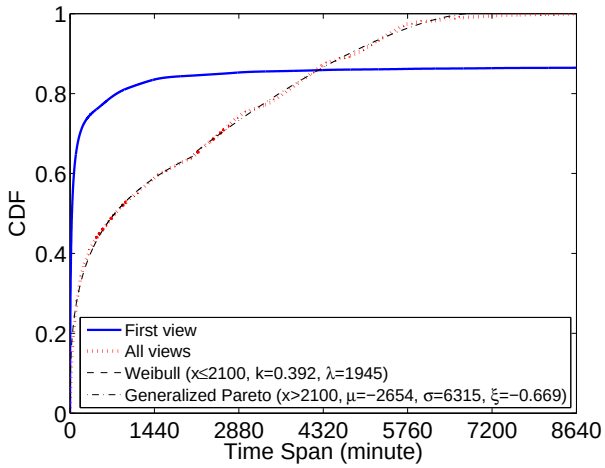


Figure 4.10: CDF of time span from share to view

only active in sharing (users active in initiating videos are likely to be SU).

It is worth emphasizing that, the analysis and modelling of user's initiating, watching, and sharing behaviour is not only important to understand user's diverse behaviour, but also crucial to facilitate the model of video spreading in the social networking services, as we will discuss in Section 4.5.

4.4 Temporal and Spatial Characteristics of Video Spreading

We now examine the temporal and spatial characteristics of video spreading from a global view, beyond the behaviour of individual users.

4.4.1 Temporal Locality

We first check the time span between sharing a video and the actual view of this shared video by the sharer's friends. We examine the sharing records that are created in the first two days, and the corresponding viewing records within 6 days. The reason we do not examine all the sharing records is that, sharing records created in the last day only have less than 24 hours to be watched, leading to unfair comparison. We define the view from the first friend that watches the video as "first view", and if a shared video has not been watched in 6 days, we set the "first view" value to 8640 (minutes of 6 days); all the views

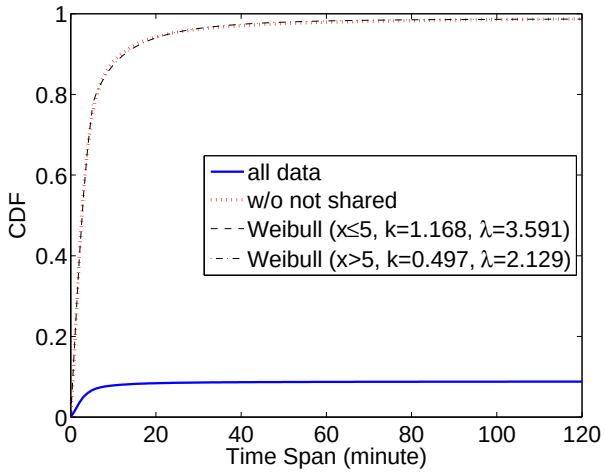


Figure 4.11: CDF of time span from view to share

by friends are defined as "all view". The respective CDFs of the time spans for both are plotted in Figure 4.10.

We observe that 13% of the shared videos will not be watched in 6 days, and for those videos that have been watched, 68% can be watched within one hour. This indicates that videos can quickly propagate to friends in the social networking service, exhibiting strong temporal locality. By examining the data of "all view", we find that only 2.6% views appearing after 4 days and less than 1% after 5 days. This implies that the life span of video spreading in the social networking service is in general of short durations, and thus our one-week dataset is suitable for the study.

We tried several common distributions to fit the curve (we only fit the "all view" which will be utilized in our model in Section 4.5), but none of them fit well. As a result, we obtained a good fit from a combined distribution with Weibull and generalized Pareto.

We then compute the time span between watching a video and sharing it, i.e., how long it takes a user to share a video to friends after clicking to watch it. We find that over 90% of views are not followed by sharing. For the rest of the records, they are clearly affected by the video length: 88% of shares are created within 10 minutes after the users started watching a video, which can be well explained that the videos shared in social networking services are mostly short user-generated contents.

Figure 4.11 plot the CDF of the time span within two hours, as well as a combined Weibull CDF fit. Note that, there are only 1.3% of shares occur after two hours. The curve

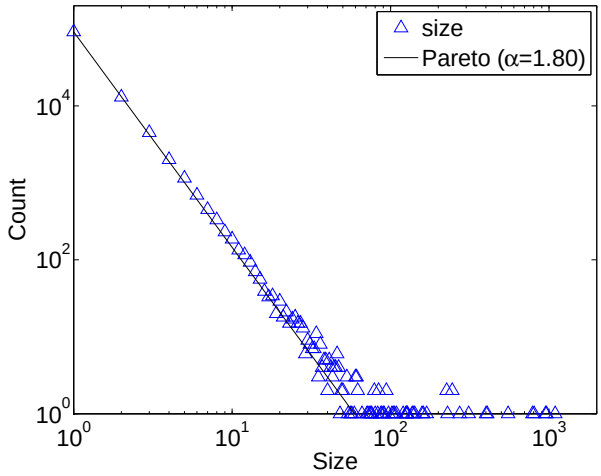


Figure 4.12: Distribution of spreading tree size

implies that not only the videos are short, but also the users tend to share them right after finishing watching (or even before finish). Again, the modelling of time spans from share to view and from view to share facilitate the video spreading model in Section 4.5.

4.4.2 Spatial Structures

We next study the spatial structures of video spreading. Consider each user that has shared a video as a node, and if user a shares a video that is previously shared by user b, then a directed edge forms from b to a. It is easy to see that the videos are spreading along a tree structure. Note that, we ignore the users that only watch the video in the spreading tree.

By associating viewing and sharing events, as discussed in Section 4.2, we obtain all the father-child relations for forming the tree structures. We use a bottom-up method to construct the propagation trees. It is possible that the entire spreading tree for one certain video may be broken into several trees because some relations that connect the tree are missing, due to that not all users will watch the video before sharing it, i.e., some users will share the videos without watching them. In this case, we estimate the relationship by connecting the two nodes that have the closest sharing time. As a result, we obtain over 23 thousand spreading trees.

We first examine the tree size and height. The former reflects the popularity of the shared video, and the latter corresponds to the maximum number of hops that the video

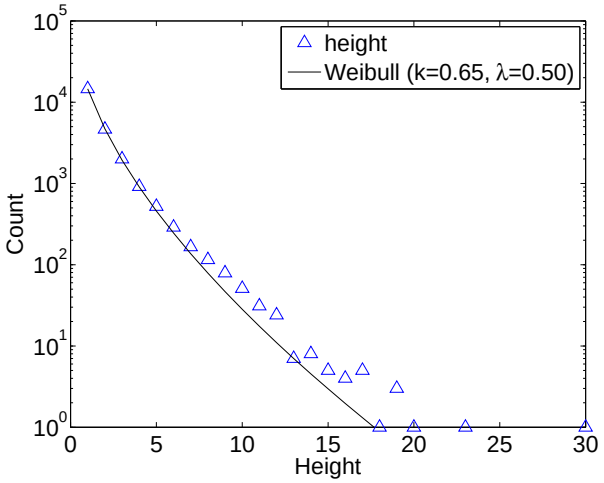


Figure 4.13: Distribution of spreading tree height

has spread, which also indicates the "liveness" of the video. As shown in Figure 4.12, not surprisingly, the distribution is long-tail scale-free, and can be well fitted by a Pareto distribution. We observe that most of the trees have small sizes, indicating most videos are spreading in small range, showing locality of social relationship; yet there also exist trees with size greater than one thousand. We find out that the most popular video has been watched by near 70 thousand users. This shows the great coverage of video spreading in the social networking services.

We define the tree height as the largest length from the root to a tree node. As shown in Figure 4.13, it can be well fitted by a Weibull distribution. This observation is quite different from other information propagation structures. For example, email forwarding have ultra-shallow trees, among which 95% are of height 2 and no trees are higher than 4 [107]. The height of the video spreading trees however can reach to 30.

We next examine the tree width, which is the maximum number of nodes at a certain depth, indicating the impact of breadth of the spreading. The distribution, shown as blue circle in Figure 4.14, is very similar to that of the tree size, with a correlation coefficient as high as 0.89. We thus wonder if the depth has impact on the number of children for each tree node, i.e., the number of re-sharing.

For better visual effect, we do not show the real data, instead, we plot the Pareto fits of the node's children count from depth 0 to 3 in Figure 4.14. From the fitted parameter, we

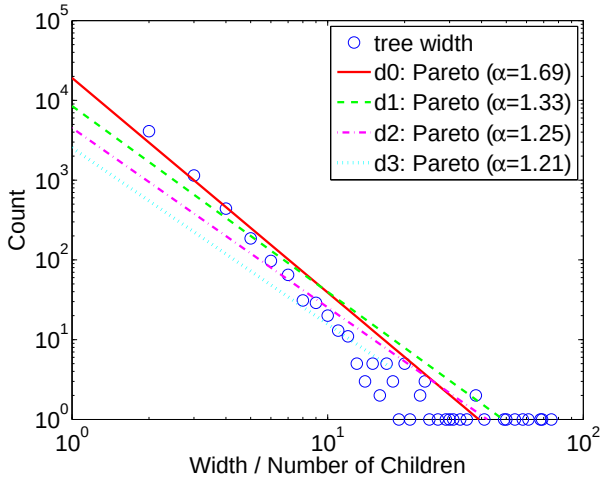


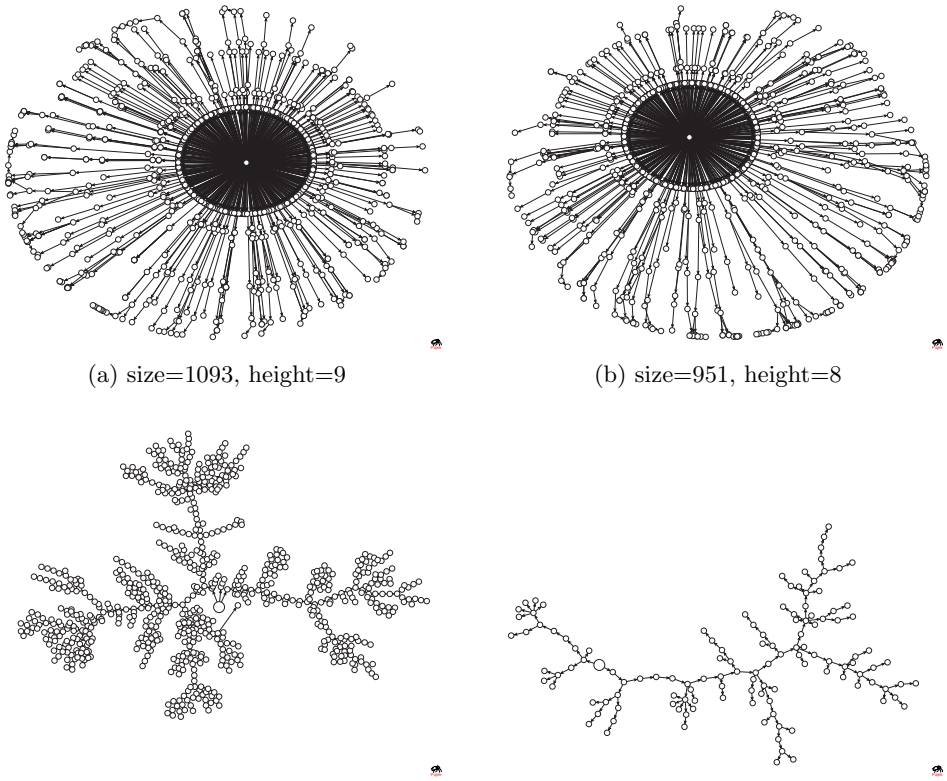
Figure 4.14: Distribution of spreading tree width

size	height	views	length	category	description
1093	9	34,531	123s	news	a father picked up daughter from school by helicopter
951	8	14,281	60s	advt.	earth hour promotion video
805	30	12,658	306s	music	charity single "Children" by Chinese stars
126	23	1,431	235s	comedy	funny lip sync video

Table 4.1: List of ten popular videos

can see that, except for the first depth, there is no much difference among the distributions of the other depths. Therefore, we can conclude that most of the users do not have and do not need to have the knowledge of their spreading stage, and whether or not to spread entirely depend on the users themselves. This is drastically different from other propagation studies, particularly on email forwarding, which exhibits strong stage dependence [107]. As such, such previous models as branching may not work for modelling video spreading.

Finally, we investigate the tree shape of the spreading of popular videos. We observe two typical spreading trees, and a visual illustration (generated by Pajek [80]) of four examples is shown in Figure 4.15: one type has moderate depth, but most of the nodes are the children of the root node, shown by Figure 4.15a and Figure 4.15b; the other type has large depths, and the number of children is not very diverse, shown by Figure 4.15c and Figure 4.15d (the root node is enlarged for better view). We further check the video content from each video URL, and list the descriptions in Table 4.1.



(a) size=1093, height=9

(b) size=951, height=8

(c) size=805, height=30

(d) size=126, height=23

Figure 4.15: Illustration of spreading trees for popular videos

This observation indicates that the video spreading trees are highly diverse, mainly due to the diverse user behaviour. Therefore, those deterministic model trying to structurally and statically capture the propagation might not be unsuitable for video spreading. Characterizing user's status evolution is thus essential to understand the video spreading.

4.5 Video Spreading Model

As mentioned above, previous models that are trying to capture the static structure of propagation are not suitable to model video spreading, due to the diverse user behaviour. An alternative method is to capture the user's status evolution, and the widely-used epidemic model is an appropriate method for modelling video spreading.

4.5.1 Epidemic Model Primer

An Epidemic model describes the transmission of communicable disease through individuals [34]. Besides in epidemiology, it has also been recently used to model computer virus infections and information propagations such as news and rumors [68, 47].

One of the classical epidemic model is the *SIR model* (Susceptible-Infectious-Recovered), first proposed by Kermack and McKendrick [62]. It considers a fixed population with three compartments: Susceptible (S), Infectious (I), and Recovered (R). The initial letters also represent the number of people in each compartment at a particular time t, that is, $S(t)$ represents the number of individuals not yet infected; $I(t)$ represents the number of individuals who have been infected and are capable of spreading the disease to those in the susceptible category; $R(t)$ represents the number of individuals who have been infected and then recovered. Given transition rate β from S to I and γ from I to R, the following equations can be derived:

$$\frac{dS}{dt} = -\beta SI, \quad \frac{dI}{dt} = \beta SI - \gamma I, \quad \frac{dR}{dt} = \gamma I.$$

4.5.2 The SI^{2}RP Model

There is a natural mapping between conventional object sharing propagation in social networks and the compartments of the SIR model. For a particular object, all the users in the social networks are Susceptible at the beginning; at a certain time, the users accessing the object are Infectious, indicating that they are able to infect others by sharing the object. They can be Recovered if they choose not to share.

From our experience of spreading videos in social networks and the investigation of user behaviour, we observe the three stages of video spreading, and the initial mapping of the SIR model. However, for video spreading, the mapping is far from being complete:

1. As we discussed earlier, most of the videos spread in relatively small ranges, covering only a small portion of the users in the entire social networks. We introduce a new compartment, Immune (Im, and thus Infectious is abbreviated to In), to indicate those users who have not watched or shared the video. Furthermore, a user can choose not to watch the received video, and possibly not participate in the spreading as well. To differentiate these users and the users in R who have watched or directly shared the video, we also categorize these users to Immune.

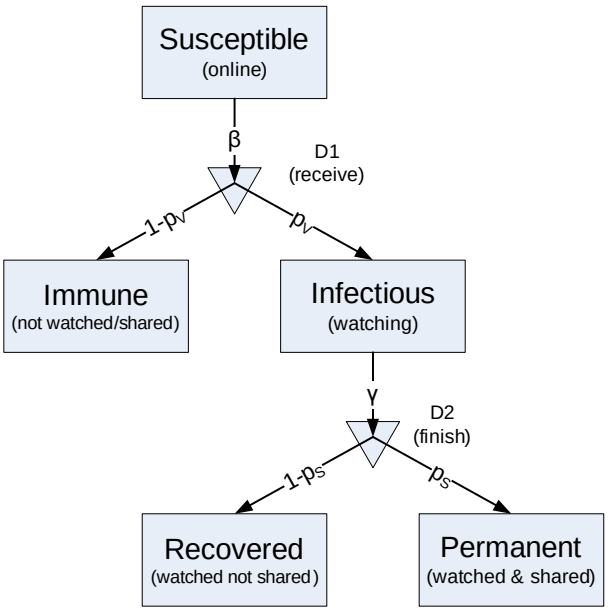


Figure 4.16: SI^2RP model

2. In the classical SIR model, the transition is time-dependent, i.e., at any time, there is a chance that the stage transits to the next one. While for video spreading in social networks, the transition of the stages depends on decisions at a certain time. For example, the user needs to choose watch or not, and share or not share. Therefore, we introduce two temporary decision stage, D1 and D2;

3. We also need to differentiate the users who have shared the video and those who have not after watching the video. Therefore, we introduce a new compartment, Permanent (P), indicating users who have shared the video, and otherwise Recovered.

The enhanced SI^2RP (Susceptible-Immune-Infectious-Recovered-Permanent) model is illustrated in Figure 4.16. Since our focus is on the spreading of video, instead of the initiation of the video, we assume the initiator is Infectious at the beginning.

The transition rate from S to D1 is β, and thus a Susceptible user will spend $1/\beta$ unit time to receive a shared video from a friend. The user then makes a decision whether or not to watch the video. If the user is not interested in it and decides not to watch or share, s/he can be considered as Immune. We denote the probability of the user watching or directly sharing the video as p_V, which will be quantified later.

If the user decides to watch the video, s/he becomes Infectious. The transition rate from I to D2 is γ, indicating that the user will spend $1/\gamma$ time to finish watching the video. The user then makes the second decision, whether or not to share the video. If the user decides not to share, s/he becomes Recovered or Permanent otherwise. The probability of a user deciding to share the video is denoted by p_S.

For easy of exposition, we assume (1) a user will make the first decision right upon receiving a friend's video share; (2) the user will make the second decision right after finishing watching the video. In addition, our measurement study on user behaviour and temporal structure facilitate with the parameters of the model. Since we also take the outlier into account, there is no overfitting issue.

The transition rate β and γ can be inferred from the measurement results in Section 4.4. Specifically, $1/\beta$, the time span from share to watch, is well fitted by a combined Weibull and Generalized Pareto distribution with CDF

$$f_{x_m,k,\lambda,\mu,\sigma,\xi}(x) = \begin{cases} 1 - e^{-(x/\lambda)^k} & x \leq x_m \\ 1 - \left(1 + \xi \cdot \frac{x-\mu}{\sigma}\right)^{-\frac{1}{\xi}} & x > x_m \end{cases}$$

where $x_m = 2100$, $k = 0.392$, $\lambda = 1945$, $\mu = -2654$, $\sigma = 6315$, and $\xi = -0.669$ from our measurement, and x is the unit time (in minute); similarly, $1/\gamma$, the time span from watch to share, is well fitted by a combined Weibull distribution with parameter $x_m = 5$, $k_1 = 1.168$, $\lambda_1 = 3.591$, $k_2 = 0.497$, and $\lambda_2 = 2.129$ from our measurement, and x is again the unit time (in minute).

4.5.3 Users Classification

We now quantify D_I, p_V, and p_S in the above model using our earlier measurement data. Note that these four probability distribution or probability characterize the behaviour of different types of users, namely, spreaders (SU), ordinary users (OU), and free-riders (FU).

- An SU initiates video shares according to distribution D_I (OU and FU do not initiate);
- A user watches videos shared by friends with probability p_V, which is based on the reception rate;
- After watching, an SU or OU shares the video with probability p_S, which is based on the share rate.

Table 4.2: Validation of SI^{2}RP model

statistic	**fit**	R^2 **of model**	R^2 **of validation**
reception rate	GPD	0.9981	0.9958
share rate	GPD	0.9966	0.9535
time to watch	Weibull + GPD	0.9990	0.9346
time to share	Weibull + Weibull	0.9991	0.9811

From the measurement in Section 4.3, we have

$$D_I(x) = \frac{1}{x^a},$$

where $a = 1.09$ and x is the rank index of the initiator. Note that $D_I(x)$ is the PDF; hence, to emulate the number of initiated videos for each initiator, a total number of initiated videos should be defined and then multiplied by the PDF.

The reception probability p_W follows a generalized Pareto distribution (GPD) with CDF

$$f_{\mu,\sigma,\xi}(x) = 1 - \left(1 + \xi \cdot \frac{x-\mu}{\sigma}\right)^{-\frac{1}{\xi}},$$

where $\mu = -0.004$, $\sigma = 0.182$, and $\xi = -0.215$ from the measurement, and x is between 0 and 1. Similarly, the share probability p_S can also be inferred from the measured share rate in Section 4.3, specifically, GPD with $\mu = -0.227$, $\sigma = 0.305$, $\xi = -0.048$.

4.5.4 Model Validation

We have ran our SI^{2}RP model multiple times to validate its accuracy. We generate synthetic traces of 1000 users participating in 1000 video spreading for 8640 minutes (6 days). We then extract a series of statistics from the traces, such as number of received, watched, shared videos for each user, time span from share to watch, and time span from watch to share. We examine these statistics with the real dataset, specifically, we compute the R^2 of the generated data and the real data. We list those goodness of fit, as well as statistical fit names and the corresponding R^2 from Section 4.3 and Section 4.4 in Table 4.2. The high values of R^2 (above 0.9) indicates that our model accurately characterizing the user behaviour in video spreading.

4.5.5 Model Discussion

We next investigate the number of each compartment, and this will give us the knowledge of the amount of video spreading in social networks. We calculate the average and standard

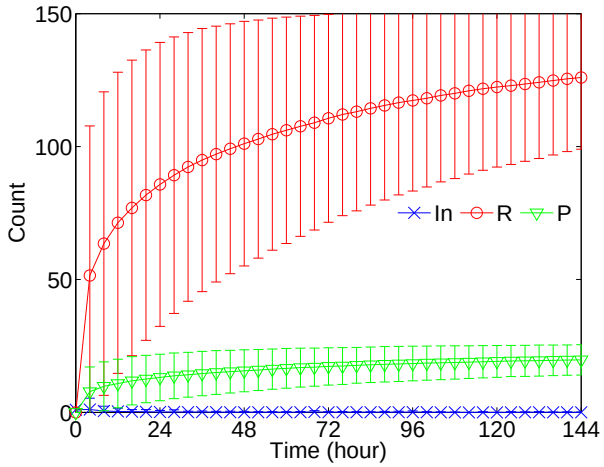


Figure 4.17: In, R, and P along time

deviations of In, R, and P in Figure 4.17. Note that, we do not show statistics of S and Im, because users in these two compartment has no impact on the network traffic. From the figure, we can see that R and P are quite diverse, this is because of the diverse user behaviour, and our stochastic model can well capture this diversity. It is also easy to observe that the growth trend of R and P are decreasing, as they are almost unchanged after 24 hours. This further confirms the temporal locality of video spreading in social networks that, a video is usually watched and shared in short time, say a few hours, and then fewer and fewer users will watch and share the video.

It is worth noting that In is quite small in the figure. This is because, first, for most of the videos, the spreading range is extremely small, and thus the number of users that have chances to watch the video is not large; second, the time span from watching a video to sharing it is generally very short, and thus In will transit to R or P very quickly. This observation, along with the temporal locality observed, indicates that for most videos spreading in social networks, the number of concurrent viewers is extremely small.

It is a trend that video sharing websites are attempting to utilize peer-to-peer technique to reduce the server workload, because users contribute resources in peer-to-peer system. The size of peer-to-peer overlay directly affect the efficiency of data delivery. Therefore, if the number of concurrent viewers is small, the traditional overlay for single video will be too small to achieve satisfying performance. This also confirms the conclusion in our previous work [21, 24]. As a result, enlarged overlay that includes a series of videos is required, and

social network can in turn help this case.

The SI^{2}RP model captures user behaviour on watching and sharing videos, as well as the latency of watching and sharing videos in the Renren network, which can be also generalized to other social network websites with certain customization. As such, the model has diverse applications; in particular, it can serve as a request generator/predictor for video accesses from a social network. Specifically, Infectious that can be derived from the model gives the number of users that are downloading and watching one particular video. The behaviour of initiators is also modelled by our measurement; thus the total number of users that are downloading all videos can be estimated. Given the video bit-rate, the traffic volume and its temporal distribution for individual users or videos and that from the entire social network can also be synthesized.

4.6 Conclusion

This chapter presented a measurement study on understanding the characteristics of video spreading in online social networking services. Based on one week of video sharing and viewing event records from the Renren network, we revealed the user behaviour from diverse aspects. We identified different types of users in video propagation and evaluate their activities. We also examined the temporal distribution during propagation as well as the typical spreading structures, revealing more details beyond stationary coverage. We further extended the conventional epidemic models to accommodate diverse types of users and their probabilistic viewing and sharing behaviour.

Our models can be applied to diverse application scenarios. Examples include a Content distribution network (CDN) or peer-to-peer video streaming system that can benefit from the temporal, spatial, and social localities found in video spreading. Our measurement on spreading tree shows that *flash crowd*, a critical challenge to existing video servers or peer-to-peer streaming system, may not be very severe with video accesses gradually spread through friends. Yet certain hub-like initiators, as identified in our measurement, may need to be carefully dealt with. Our SI^{2}RP model further provides an valuable tool to predict the video request from a social network, thus helping with server load provisioning and balancing. It may also facilitate video content providers migrating to their services to a cloud platform, through effectively forecasting traffic demands for elastically leasing resources.

Chapter 5

Coordinate Live Streaming and Storage Sharing

The recently emerged video sharing services, online social networking services, as well as the pervasive wireless mobile network services, have drastically changed the content distribution landscape. Today video sharing services such as YouTube allow any user to be a content provider, generating enormous amount of video contents that are quickly and extensively spreading on the Internet through social networking services such as Facebook and Twitter.

Unfortunately, the existing video sharing websites are facing critical server bottlenecks and the surges created by the social networking users would make the situation even worse. To better understand the challenges and opportunities therein, besides our study on video spreading in Chapter 4, we also investigate users' social behavior and personal preference of online video sharing from a user questionnaire survey. Our data analysis, presenting in Section 5.3, reveals an interesting coexistence of live streaming and storage sharing, and that the users are generally more interested in watching their friend's videos. It further suggests that even though the traffic is significant, most users are willing to share their resources to assist others, implying user collaboration is a rationale choice in this context.

In this chapter, we present COOLS (Coordinated Live Streaming and Storage Sharing), a system for efficient peer-to-peer posting of user-generated videos. The overview and dtails of the design are presented in Section 5.4 and Section 5.5, respectively. Through a novel ID code design that embeds nodes' locations in an overlay, COOLS leverages stable storage users and yet inherently prioritizes living streaming flows. We also present the improvement

of the basic overlay design in Section 5.6. In Section 5.7, the evaluation results show that, as compared with other state-of-the-art solutions, COOLS successfully takes advantage of the coexistence of live streaming and storage sharing, providing better scalability, robustness, and streaming quality. Finally, we conclude this chapter in Section 5.8.

5.1 Introduction

Recently, video sharing services and social networking services have become highly integrated. YouTube enables automatic post on such social networking websites as Facebook and Twitter based on user's options, and users can also share interested videos on their social networking webpages. In fact, YouTube had been brought directly into Google+ [53], making it easier to watch and share [77]. YouTube reveals that 500 years of YouTube video are watched every day on Facebook, and over 700 YouTube videos are shared on Twitter each minute; moreover, the YouTube player is embedded across tens of millions of websites [125]. The videos are spreading in the social networks, bringing significant challenges not only to the social media website management, but also to the network traffic engineering.

Connecting people through cascaded relations, social media spreads information much faster and more extensively than conventional web portals or newsgroup services. Together with the pervasive penetration of wireless mobile networks and advanced devices (e.g., smartphones and tablets), TV-quality video contents can now be truly generated and accessed anywhere, at any time, and by any person. It is reported that YouTube mobile gets over 600 million views a day, and traffic from mobile devices tripled in 2011 [125].

Unfortunately, the sheer and ever increasing data volume, the broader coverage, and longer access durations of video objects also present significant challenges than other types of objects, not only to the social media website management, but also to the network traffic engineering and to the resource provisioning of external video sites. It is known that YouTube-like sites are facing critical server bottlenecks [17], and the surges created by the social networking users would only make the situation worse. In fact, even the text-based Twitter has encountered system-wide outages during some critical events, e.g., the Obama's inauguration [2] and Michael Jackson's tragical death [99]. While peer-to-peer has long been advocated as a solution for TV or movie content streaming, it remains unclear whether it is doable for the user-generated videos with independent asynchronous viewers.

From the measurement study we have conducted in Chapter 4, we found that social

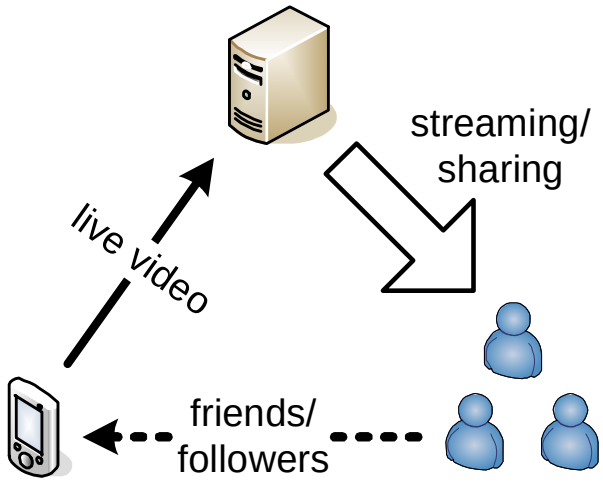


Figure 5.1: Application scenario

networking users watch and shared a great amount of videos. They watch a good portion of videos posted by friends, and thus together with the large amount of videos posting in the social network, this implies that client/server might suffer from lack of scalability. We also found that the interval of posting and watching video is relatively short, indicating that there is a flash-crowd after the video is posted.

To better understand the challenges and opportunities therein, we have also conducted a user questionnaire survey on their personal preference and social interest of Internet video sharing, to directly learn user behaviour. The survey result reveals an interesting coexistence of live streaming and storage sharing; that is, upon on receiving a video post, social networking users can watch the video immediately, or download and then watch later.

As illustrated in Figure 5.1, users can use the built-in camera or mobile devices to record video, and simultaneously send the live video to a server, such as YouTube and Ustream. Through posting function of social networking services, the server can broadcast the live video to the user's friends, who can be either wired Internet users or mobile users. Upon receiving the video post, a friend has three options:

1. a friend can choose to watch the live video, and thus the requirement of streaming quality, such as, startup latency and playback continuity, should be satisfied;

2. a friend can choose not to watch the live video, but s/he can download the video and expect to watch it later. Hence, such a user is considered **delay-tolerant**. In addition,

the user may also switch to the first option at some time during the live streaming;

3. a friend shows no interest in the video. In this case, if such a user does not want to watch the video now or later, s/he may not want to share the resources with other uploader's friends, either.

The coexistence clearly makes a system design more complicated. It however also suggests that semi-synchronized user for video streaming may reach a critical mass for collaborative streaming, and that the users downloading the video are considered relatively stable, and thus could be leveraged to combat node churns. More importantly, our survey reveals that most of the users are willing to share their resource to assist others with close relations. Also, although people are not fully satisfied with the playback quality provided by most of the video streaming services, their concern is more about the content of the video, which largely determines the watching duration. Consequently, if their friends upload videos, they will be more interested and likely watch more of the video. All these features imply that collaborative peer-to-peer is a rationale and promising choice in this scenario.

In this chapter, we present COOLS (Coordinated Live Streaming and Storage Sharing), a system for efficient peer-to-peer posting of user-generated videos. Through a novel ID code design that embeds nodes' locations in a tree overlay, COOLS leverages stable storage users and yet inherently prioritizes living streaming flows with short startup delay. It also gracefully accommodates users' switch from the second option (storage) to the first one (live streaming), as well as node dynamics. We also improve our overlay tree to achieve better efficiency and robustness. The evaluation results show that, as compared to other state-of-the-art solutions, COOLS successfully takes advantage of the coexistence of live streaming and storage sharing, providing better scalability, robustness, and streaming quality.

5.2 A Brief Revisit of Video Spreading in Social Networks

In Chapter 4, we have conducted a measurement study on video spreading in the social networking service, based on one week of video sharing and viewing traces from Renren.

When a social networking user posts a video, her/his friends will be notified in the news feed on the social networking website. Web users will only notice the feed when log on to the website, while mobile users can be immediately notified. Different from text or images that can be instantly viewed, a posted video will not be really watched until the recipient clicks

the link. The user can also further share the video, so that the video post will spread in the social network. We examined the number of videos a user has watched and shared. Not surprisingly, both distributions are highly skewed, displaying long-tail scale-free property, as shown in Figure 4.3 and Figure 4.6.

Besides the number of videos watched for each user, we also considered the number of the user's received video posts from friends. We computed the ratio of the number of watched videos and the number of received video posts from friends, defined as the *reception rate* for each user, and the result is shown in Figure 4.4. On average, users watch 16% of videos shared from friends. Although this number is not large for the individual user, considering the huge number of users and posted videos in the entire social network (refer to Figure 4.3 and Figure 4.6), we can observe the great number of participants for the video posting and the great number of video posts. This suggests us the client/server architecture will suffer from huge amount of usage, and also peer-to-peer delivery mechanism is a possible solution.

We also studied the time span between posting a video and the actual view of this posted video by friends. The result is shown in Figure 4.10. We observe that except 13% of the posted videos that will not be watched in 6 days, 68% of the rest can be watched within one hour. This indicates that videos can quickly spread to friends in the social network, exhibiting strong temporal locality. We also concluded that, although there are some friends accessing the video after a while, most accesses occur in a short time after the video is posted, displaying a flash-crowd properties. From this conclusion, we can simplify our application scenario that we assume all the users join the system at the beginning, which is different from the conventional streaming scenario.

5.3 A User Questionnaire Survey

Most of the existing studies on video sharing services measured the log traces and data crawled from the webpages to derive user-related statistics. Trying to further and directly understand the Internet users' preference and social interests on viewing and sharing online videos, we created a web survey and invited worldwide people to fill in. The survey contains a series of single-choice questions plus several questions on insensitive personal information. The details of the survey is shown in Appendix B.

As a result, 117 people have participated in the survey. 69 are from North America, 39 are

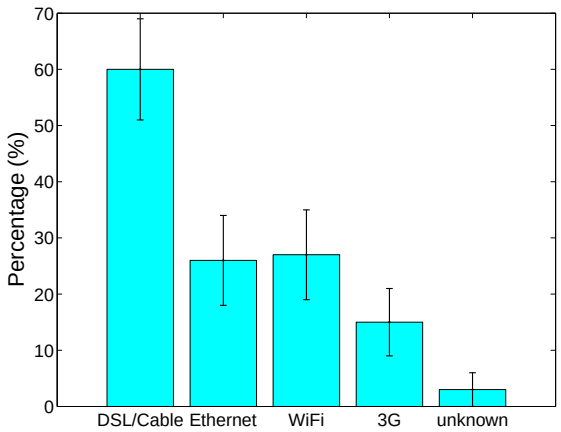


Figure 5.2: Breakdown of user's network connection with confidence interval (confidence level being 95%)

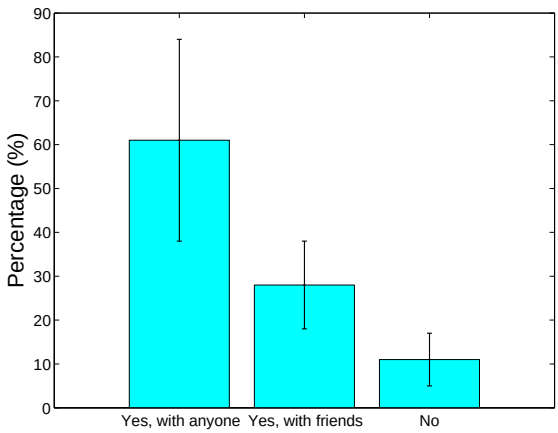


Figure 5.3: Breakdown of user's willingness of contribute with confidence interval (confidence level being 95%)

from Asia, and 8 are from Europe, with various network connections, refer to Figure 5.2.[1] $(91 \pm 5)\%$ of them are of ages 19-30, which is exact the core generation of the Web 2.0 applications users.

We now summarize the key observations from the survey results. We find that $(62 \pm 9)\%$ participants usually leave a video streaming session after selecting it and return after a while, rather than stay and wait for the startup, and $(84 \pm 7)\%$ of them consider playback quality as the key factor of this behaviour. In fact, only $(55 \pm 9)\%$ and $(61 \pm 9)\%$ are satisfied with the startup latency and playback continuity, respectively. Then, users come back in a certain returning time. In terms of the returning time, some users consider the video length, and some consider the absolute waiting time. Considering the video length, $(46 \pm 9)\%$ of the users come back after a quarter of the video has been downloaded, and $(68 \pm 8)\%$ come back after half of the video has been downloaded; $(22 \pm 8)\%$ of them wait until the entire video is downloaded. While regardless the video length, $(69 \pm 8)\%$ of the users spend less than 5 minutes for waiting, and no one will wait for more than 30 minutes. In short, for the same video content, **viewers of streaming and that of store-and-play both exist**.

Second, the survey asked users if they are willing to share their resources while streaming and downloading, regardless any particular implementation (i.e., browser add-on, specific software). The result, as shown in Figure 5.3, is gratified that only $(11 \pm 6)\%$ of users do

[1]Since this is a multiple choice question, the summation of the percentage is greater than 1.

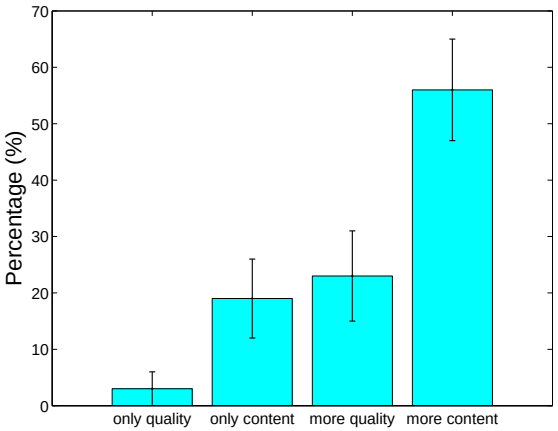


Figure 5.4: Breakdown of user's concern on videos with confidence interval (confidence level being 95%)

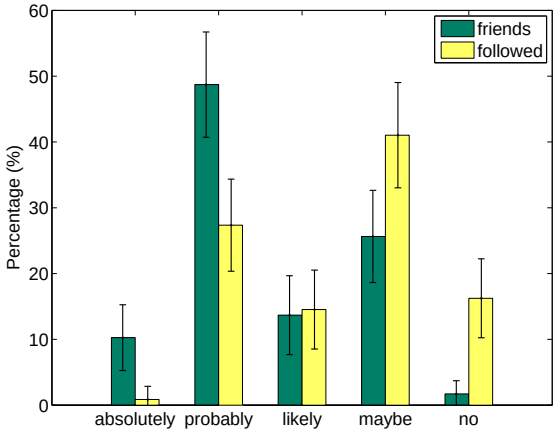


Figure 5.5: Comparison of the possibility of watching the entire video with confidence interval (confidence level being 95%)

not want to contribute. $(61 \pm 23)\%$ of the users do not care who they are sharing with, and $(28 \pm 10)\%$ users only want to share the resource with close relations, e.g., friends in Facebook and mutual followers in Twitter.

Third, only $(57 \pm 9)\%$ of the people tend to watch the entire video in general. Not surprisingly, this behaviour is affected by the video content, as three quarters of users concern more about the video content than the playback quality, as shown in Figure 5.4. Interestingly, when a video is uploaded by a friend, a user are more likely to watch more of the video. Figure 5.5 shows the comparison of the possibility of watching the entire video, uploaded by a friend and someone the user just followed. The figure clearly implies that a video will be watched more, if it is uploaded by a user with closer relation. Unfortunately, we did not get such data on the case of watching video uploaded by a total stranger, but we believe the number will be much lower than that of someone followed. In short, this observation, together with that most of the users are willing to share their resources, indicates that user collaboration is rationale in this scenario.

5.4 COOLS System Overview

5.4.1 Streaming User and Storage User

As suggested by the survey, there exist two types of friends interested in the posted video, namely, *streaming users* and *storage users*. The streaming users expect to watch the video immediately, and the storage users will download and watch the video at a different time, due to the presence of other concurrent events. When the storage users start to watch, they can either watch from the beginning or watch the current live stream, given that the live broadcast is not finished. We ignore the users that are not interested in the video.

The streaming users might stop watching after a while if they find the video is out of their interest, even though the video is posted by friends. Users leaving the system causes dynamic and can affect the data delivery. On the other hand, the storage users that are downloading the video asynchronously do not have the concern of interest nor playback quality, until they start to watch the video, and we assume the users will not leave the system. Hence such users are considered relatively stable, though they could switch their options during downloading the video, and become streaming users in that case. Therefore, our design principle is to leverage the stable storage users to combat node churns (i.e., node dynamic behaviour such as joining and leaving) in the data delivery system.

5.4.2 Overlay Tree

Considering the above factors, we advocate a tree overlay design for video posting. It is known that a tree overlay with data push is more efficient than a mesh overlay with data pull, but maintaining the tree with node churns is a daunting task. Fortunately, the existence of storage users implies that their churns are much less frequent than the traditional live streaming, which can thus be strategically placed to improve the robustness of a tree overlay.

To efficiently coordinate the two types of users, we implement a labeled tree that embeds node locations in the overlay. For ease of exposition, we explain it with a binary tree and an example is given in Figure 5.6. Each node is assigned an ID, represented by a series of binary code. The two children of the root node (the source) have ID 0 and 1, respectively. For a given node, its left child's ID is the node's ID appended by a 0, and the right child's ID is that appended by a 1. As such, the ID embeds the location of a node and also that of its all ancestors. Moreover, the number of digits (*length*) indicates its depth in the tree.

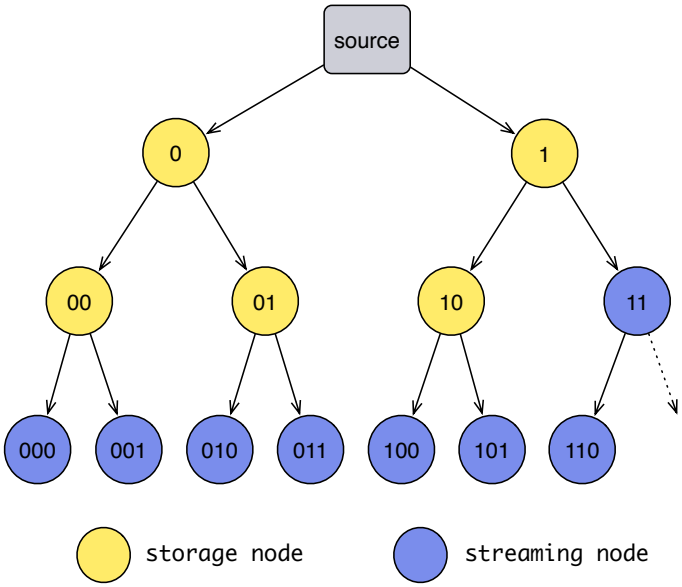


Figure 5.6: Example of overlay tree with ID

We define a partial order of the ID: if two IDs are of identical length, the one with greater value is considered greater (for example, 010 is greater than 001); otherwise, the longer ID is greater (for example, 000 is greater than 11).

We also define an *increment* operation of the ID: if not all the bits of the ID is 1, an increment operation will increase the ID value by 1; otherwise, the length of ID will be increased by 1 and all the bits are set to 0. We also denote the value of an ID increaded by 1 as the *next value* of the ID. The operation of *decrement* can be defined similarly, while in the opposite way.

Note that, we use a binary tree for easy exposition here and in the following section. The overlay tree can be extended with more children, as we will discuss in Section 5.6.

Since the storage nodes are relatively more stable, we expect that the storage nodes are placed at more critical locations of the tree, that is, close to the source. In other words, the storage nodes' IDs are smaller than that of streaming nodes after the tree is stabilized. Figure 5.6 shows the organization of two types of nodes in the overlay tree. We will detail the construction and maintenance of the overlay in the next section, particularly on both achieving robustness with storage users and minimizing delay for streaming users.

5.5 COOLS Design Details

5.5.1 Overlay Construction

Creating Storage Tree and Streaming Tree

As mentioned, the storage nodes are expected to be close to the source. However, we also need to guarantee short startup latency for the streaming nodes, which requires them to be close to the source as well. Fortunately, since the storage users are delay-tolerant, the dilemma can be eliminated by prioritizing the streaming nodes in the initial stage.

Specifically, COOLS first constructs two trees, one contains all the streaming nodes, referred to as *streaming tree*, and the other contains all the storage nodes, referred to as *storage tree*. The source only delivers data in streaming tree at the beginning. After the two trees are established, and the streaming nodes have buffered enough data to avoid timeout, the two trees will be merged to one final overlay tree.

The source records the current maximum ID of each tree. To construct the two trees, the source adds nodes to the corresponding trees sequentially. A newly added node will be assigned an ID as the next value of the maximum ID. The node thus knows its parent's location by checking the prefix of its own ID. If the source has enough children (2 in this case), it will provide the address of one of its children whose ID is the same as the first digit of the new node's ID, that is, which branch should the new node go; otherwise, the new node becomes one of the source's children.

It is worth noting that each node only keeps the local information such as the network address of the parent, two children, and the source, while the source only keeps the information of the four depth-1 children as well as the two maximum IDs. Therefore, the COOLS design shows good scalability, as the required information is independent on the number of nodes in the system.

Merging Tree and Node Promotion

At the beginning, the source dedicates to the streaming tree. When the streaming nodes have buffered enough data for playing back and avoiding timeout, the source starts to push video data to the storage tree. In the meantime, the source stops pushing data to the streaming tree and notifies the two depth-1 streaming tree children to connect to the parents, which are found by the source utilizing the ID code design. Since the streaming nodes have sufficient

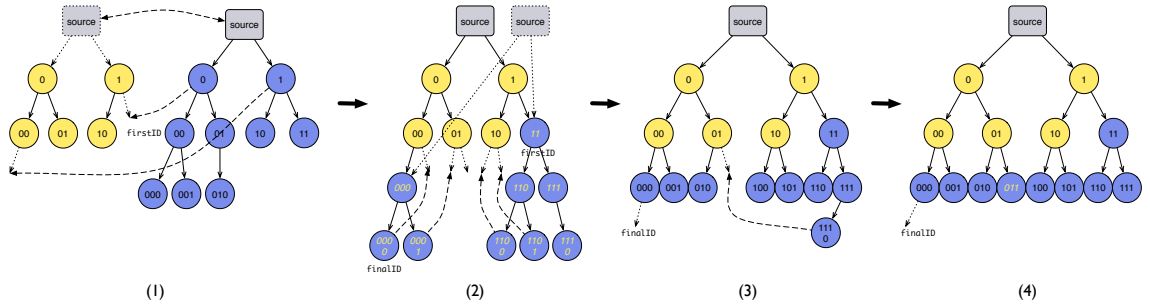


Figure 5.7: Example of overlay construction: creating, merging and promotion

amount of the video data, they will join the storage tree seamlessly without interruption of playback. The first step in Figure 5.7 shows the procedure of merging the two trees.

To simplify our exposition, we denote the next value of the current maximum ID of the storage tree as $firstID$, as it will be the first ID of a streaming node after the two trees have been merged. Therefore, the new ID of the left streaming child node changes to $firstID$, and the right streaming child node is assigned the next value of $firstID$. It is worth noting that the value of $firstID$ is not changed.

In addition, the source also computes a potential maximum ID based on the values of the two original maximum IDs, denoted as $finalID$, e.g., 0000 in this case shown in Figure 5.7. Then the source disseminates this value throughout the tree. After the two trees have been merged, the overlay tree is probably not a complete tree, as some streaming nodes may locate deeper than expected, based on the $finalID$. These nodes are in an *unsteady state*, e.g., node 0000, 0001, 1100, 1101 and 1110 in the second step of Figure 5.7. Some leaf storage nodes are also unsteady if they should have children but do not have yet, also based on $finalID$, e.g., nodes 00, 01 and 10. Other nodes are in a *steady state*. Since most of the unsteady streaming nodes are moving upwards, we call this procedure as *node promotion*.

The unsteady nodes send control messages toward the source. Specifically, if the node finds out that its ID is no smaller than $finalID$, it will send a *promotion message*; if its potential children's ID is smaller than $finalID$ but do not have any child, the node will send a *child requiring message*. A rendezvous node (not necessary the source) receiving such messages matches them, and notifies the two senders to connect with each other. For example, in Figure 5.7, node 00 matches itself with node 0000, node 0 matches node 01 with node 0001, node 1 matches node 10 with nodes 1100 and 1101, and the source matches node

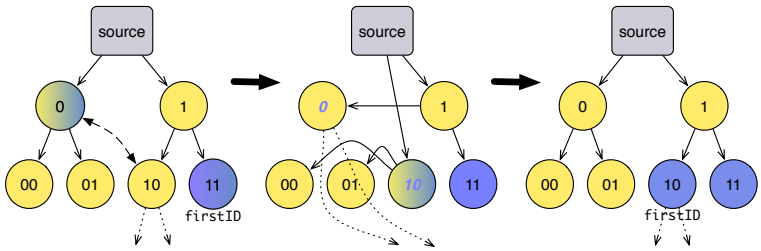


Figure 5.8: Example of node demotion

01 with node 1110.

We now mathematically analyze the process of node promotion. Suppose the heights of the original two trees are H_l and H_s, respectively. To merge and promote, in the worst case, all the promotion and child requiring messages are matched at the source. Thus in each round, the nodes in the lowest depth send promotion message and get matched, which takes $(H_s + H'_l)$ unit time, where H'_l is initially H_l and decreases by 1 each round. The tree's height will eventually become H, and all the nodes between depth H_s and depth H are streaming node. Hence there will be $(H_l + H_s - H)$ rounds. For a complete tree, all the three heights are bounded by $O(\log N)$, and the time to complete the promotion is therefore bounded by $O((\log N)^2)$, where N is the total number of nodes in the system.

5.5.2 Handling Node Dynamics

A storage user may finish her/his current event and start to watch the live video after a while, becoming a streaming node. In addition, there is also possibility that a streaming user finds the video out of interest, and thus stops watching and leaves the system. Given that the users are watching more of the entire videos that are uploaded by their friends, such events are relatively rare in our application scenario, yet proper handling of node dynamics is still necessary, as addressed below.

Node Demotion

For switching from storage node to streaming node, the node needs to be demoted in the tree. It is worth clarifying that the demotion does not degrade the playback quality; instead, it just lowers the depth in the tree, since the node becomes more possible to leave the overlay.

Figure 5.8 shows an example of the demotion process. Supposing user 0 starts to watch

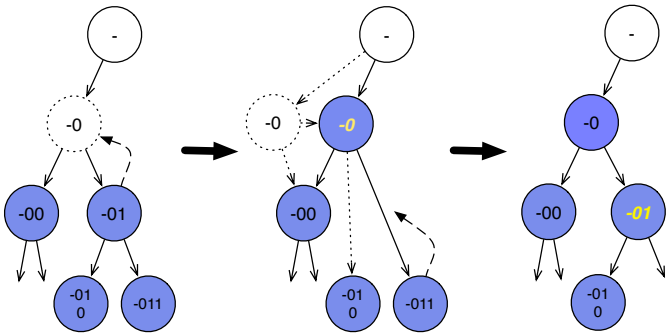


Figure 5.9: Example of node leaving

the live video, it first informs the source and gets the value of $firstID$ from the source, and the source then decreases the value of $firstID$ by one, because one storage node is switching to streaming node. In order to minimize the overhead, only the demoting node (node 0) and the last storage node (node 10) perform a swap, so that node 0 becomes the first streaming node. Then the two nodes exchange the IDs, as well as their connections with children and parents. The demotion is then completed. Since the demoting node has already downloaded sufficient data, it can immediately start watching without any startup delay. Clearly, the demotion does not affect other nodes' playback.

Node Leaving and Crash

A streaming user may stop watching after finding that the video, though uploaded by a friend, is out of interest. Node crash is also possible. Storage nodes however will not leave the system unless crash.

When a node gracefully leaves the system, it notifies the source, parent, and children about its current information of the connections. When the source receives the notification, it will compute a new $finalID$ and disseminate the new value in the overlay. To simplify the process, all the right child nodes along the path are promoted, and all the left child nodes remain unchanged. An example of node leaving is shown in Figure 5.9.

When a node crashes, neither the source nor its connected nodes will be notified. With our ID design, the crashed node's children can quickly locate their grandparent, and the grandparent thus can know the connection information of its grandchildren. The source is also notified by these children. Then these nodes repair the tree as if the crashed node

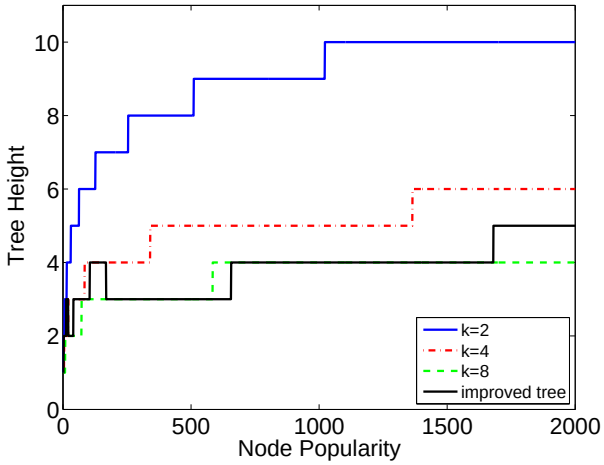


Figure 5.10: Tree height against node count

leaves gracefully.

It is worth noting that the crash of a storage node may cause that the new tree violates the requirement that all the storage nodes' IDs should be smaller than that of the streaming nodes, if the crashed storage node's right child is a streaming node. This can be addressed by the source through computing a new $firstID$ value, and switching the last storage node and the streaming node that should be demoted.

5.6 Improving COOLS Overlay Tree

In the basic COOLS overlay tree exposited in Section 5.4, each node has only two children. The benefit is that the node can devote more bandwidth to each children, and the overlay structure is easy to implement. However, the tree height can be too high if there are a large number of nodes. Particularly, the tree height is calculated as $\lfloor \log_k((N+1)\cdot(k-1)) \rfloor$, where N is the total number of nodes in the system (considering the root is the server and does not count as a node), and k is the maximum number of children the tree node has. Blue solid line in Figure 5.10 shows the tree height as a factor of the total number of nodes for a binary tree. As shown, when there are 1000 nodes, the tree height can reach to 10, making the tree vulnerable to node dynamics.

On the other hand, even though increasing the number of children for each node can reduce the height, shown as the two broken lines in Figure 5.10 for $k = 4$ and $k = 8$

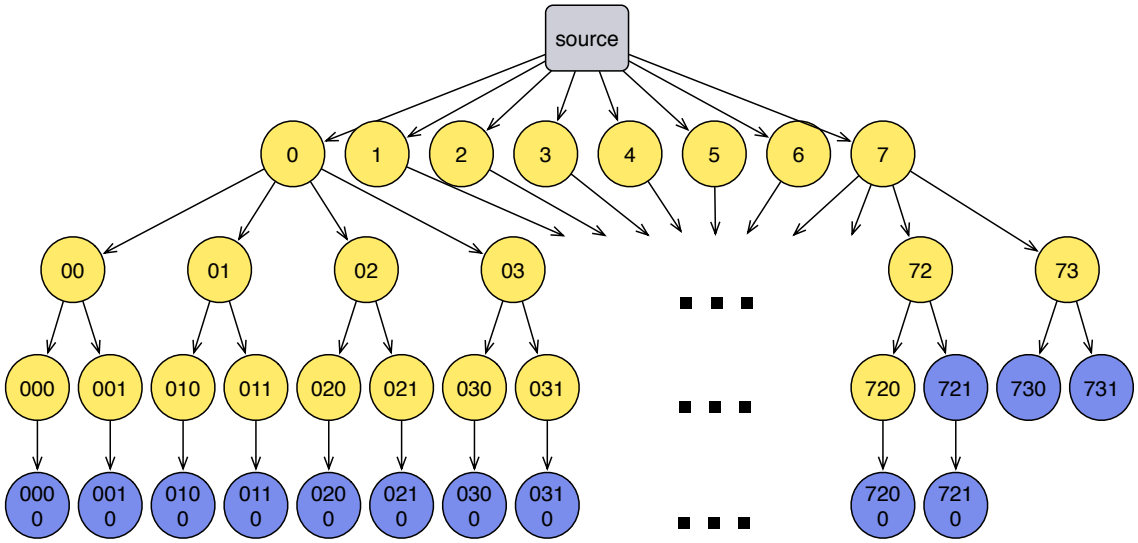


Figure 5.11: Example of improved overlay tree

respectively, simply doing so will lead to another problem: the number of nodes in each depth is growing linearly, and that will burden the nodes that are close to the root. Therefore, we expect a novel tree structure, in which both the tree height and the number of nodes in each depth grow sub-linearly.

We present a novel improved overlay tree structure: if the root node has 2^k children ($k \geq 0$), then the nodes at depth i have at most 2^{k-i} children, and the tree height is no greater than $k+1$. Figure 5.11 shows an example of a part of an overlay tree with height of 4. In the example, the root node has 8 children, the nodes at depth 1 have 4 children, the nodes at depth 2 have 2 children, and so on. Given that a complete tree in which the root node has 2^k children, there are 2^k nodes at depth 1, and $2^k \cdot 2^{k-1}$ nodes at depth 2, and so on. Thus the number of nodes in depth i is at most

$$N_i = \prod_{j=1}^{i} 2^{k-j+1}.$$

And the total number of nodes (excluding root) in the tree can be calculated as

$$\sum_{i=1}^{k} N_i.$$

Note that, there is possibility that a greater N leads to a shorter tree. To understand this, we assume a complete tree in which the root has 4 ($k = 2$) children, and thus there

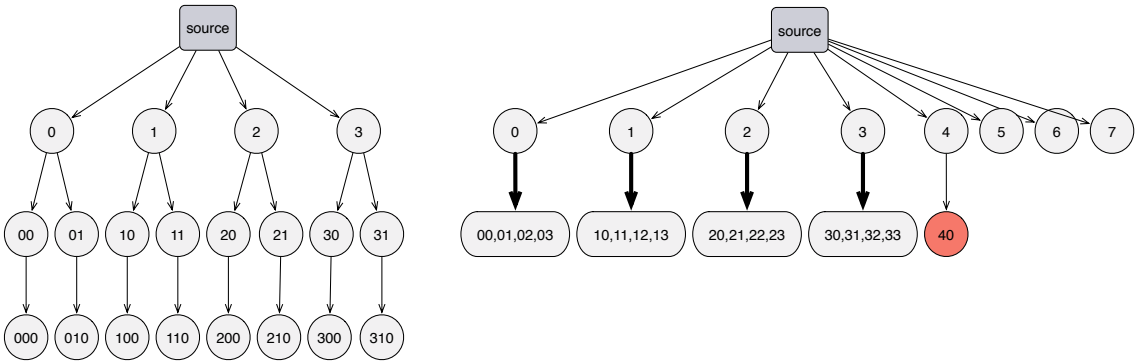


Figure 5.12: Example of greater number of nodes leading to shorter overlay tree

are at most 20 nodes and the tree height is 3 ($4 + 4 \cdot 2 + 4 \cdot 2 \cdot 1 = 20$), shown as the left tree in Figure 5.12. If there are 21 nodes, the root node has to have 8 children to satisfy the requirement (k becomes 3). As a result, the new tree's height becomes 2, shown as the right tree in Figure 5.12. Yet the new tree's height can be at most 4, and the complete tree can have at most $8+8\cdot4+8\cdot4\cdot2+8\cdot4\cdot2\cdot1 = 168$ nodes. We plot the relationship between the total number of nodes and the tree height as the black solid line in Figure 5.10. We can see that, the tree height can stay below 5 for a large number of nodes.

Since the number of nodes at depth i is $\prod_{j=1}^{i} 2^{k-j+1}$, and that at depth $(i-1)$ is then $\prod_{j=1}^{i-1} 2^{k-j+1}$, and thus the growth factor at depth i is 2^{k-i+1}. Therefore we can see, as i increases, the factor decreases, and thus the number of nodes at each depth is increasing sub-linearly, which satisfies our requirement.

Accordingly, to facilitate with the improved overlay tree, the node ID is no longer base 2 number represented by 0 and 1. In particular, at depth i of an overlay tree with height k, the ID is base 2^{k-1}, which is the number of nodes at depth 1. The tree operations (e.g., construction, demotion, and leaving) can also be adapted with marginal modification.

5.7 Performance Evaluation

5.7.1 Simulation Settings

We next present our evaluation for COOLS. In our evaluation, we use the following typical metrics, which together reflect the quality of service experienced by end users and the system performance.

Startup delay It is the time taken by a node between its request of joining the overlay and receiving enough data blocks to start playing back;

Data loss rate It is defined as the fraction of the data blocks that have missed their playback deadlines;

Control overhead It is size of the control messages sent by tree node.

To compare, we have also implemented Chunkyspread-like [105] and CoolStreaming-like [126] overlays. As a typical tree-based multicast algorithm, Chunkyspread is unstructured, using multiple trees to balance load among nodes. It also reacts quickly to membership changes and scales well with low overhead. On the other hand, CoolStreaming is a typical mesh-based data-driven overlay network for live video streaming, in which every node periodically exchanges data availability information with a set of partners, and retrieves unavailable data from partners. The design is not only efficient but also robust and resilient, and more importantly, CoolStreaming is scalable with bounded delay.

We simulate $N = 1000$ overlay nodes, which is a typical popular video overlay size. The playback will not start until the user has obtained sufficient data (10 seconds of video data). We run the simulation 100 times for each overlay to get the average results. The survey results in Section 5.3 are applied to simulate the session setting and the node dynamics (survey results are revisited in the parentheses):

1. the session length is set to $L = 1800$ seconds and each data block is of one-second video data (no user will wait more than 30 minutes);

2. 600 nodes are storage nodes at the beginning (62% of users will leave and return); A storage node switches to streaming nodes with a probability p_1 at time t_i:

$$p_1 = \begin{cases} 1/(60/0.3 - t_i) & t_i < 60 \\ 1/(240/0.4 \cdot (1 - 0.3) - (t_i - 60)) & 60 \leq t_i < 300 \\ 1/(300/0.2 \cdot (1 - 0.3 - 0.4) - (t_i - 300)) & 300 \leq t_i < 600 \\ 1/(1800 - t_i) & 600 \leq t_i < 1800 \end{cases}$$

   (among the users, 30% return within 60 seconds, 40% return after 60 seconds but before 300 seconds, 20% return after 300 seconds but before 600 seconds, 10% return after 600 seconds);

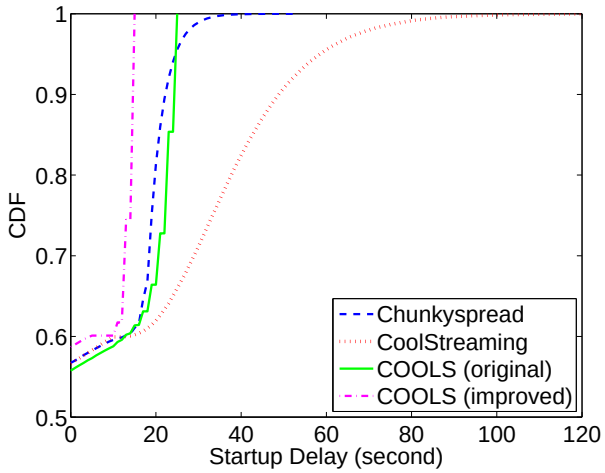


Figure 5.13: CDF of startup delay

3. A streaming node leaves the system with a probability of p_2 at time t_i:

$$p_2 = \begin{cases} 0 & t_i < 450 \\ 1/(450/0.13 - (t_i - 450)) & 450 \leq t_i < 900 \\ 1/(450/0.15 * (1 - 0.13) - (t_i - 900)) & 900 \leq t_i < 1350 \\ 1/(450/0.16 * (1 - 0.13 - 0.15) - (t_i - 1350)) & 1350 \leq t_i < 1800 \end{cases}$$

(among the users, 13% leave after watching 1/4 of the video/session, 15% leave after 1/2, 16% leave after 3/4, and 56% watch the entire video).

5.7.2 Evaluation Results

Figure 5.13 shows the cumulative distribution function (CDF) of the startup delay. Because the storage nodes have already buffered some video data before switching to streaming node, there are nearly 60% nodes having no startup delay for all the solutions. For the streaming nodes from the beginning, the mesh-based solution performs worst, because it needs a longer time to search and request for partners. The pure tree-based solution and the original COOLS perform similarly, as most of the nodes need 20 seconds to startup, but a small portion of nodes need much longer time to startup in the pure tree-based solution. This is because our COOLS solution is aware of the coexistence of the two types of nodes and explicitly prioritizes the service to the streaming nodes at the beginning. Since the improved COOLS overlay is shallower, nodes needs shorter time to startup than the original COOLS.

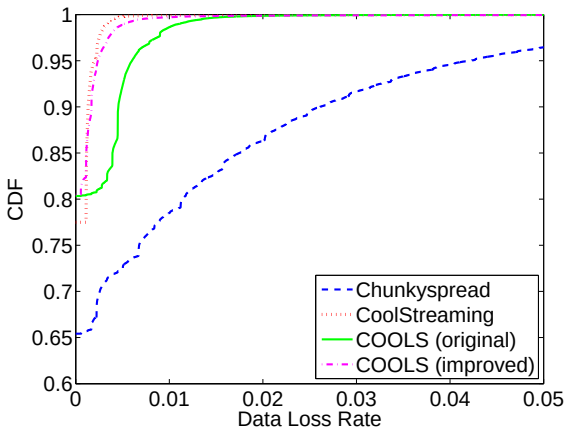


Figure 5.14: CDF of data loss rate

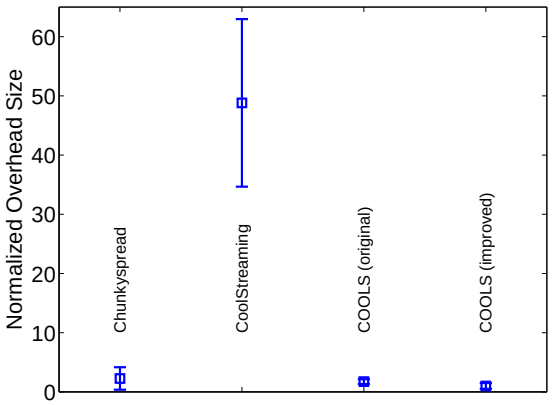


Figure 5.15: Comparison of overhead size

Figure 5.14 shows the CDF of the data loss rate. The mesh-based solution performs the best, because it is pull-based and thus is resilient to the node dynamics. On the other hand, the pure tree-based solution performs the worst, because the tree overlay is prone to suffer from the node dynamics, as there are more than 20% nodes have lost more than 1% data. Although our COOLS solution is also tree-based, through differentiating and leveraging the two types of nodes, specifically, placing stable storage nodes closer to the root, it performs much better than the pure tree-based solution. By further improving the overlay structure, the improved COOLS can achieve the performance of the mesh-based solution.

Finally, Figure 5.15 compares the control message overhead of the four solutions. Not surprisingly, with data pull, the mesh-based solution suffers from much higher overhead than the others, as it has nearly 50 times larger overhead size. The original COOLS solution has almost the same overhead as the pure tree-based solution, while improved COOLS solution has slightly less overhead. This is because (1) by node ID design, the tree overlay is well-structured, and (2) the nodes that are close to the root are less dynamic, therefore, the nodes spend less overhead on maintaining the overlay.

5.8 Conclusion

This chapter presented Coordinated Live Streaming and Storage Sharing (COOLS). The COOLS design was motivated by a real-trace measurement study and a questionnaire survey on user behaviour of Internet video sharing, which reveals a coexistence of live streaming

and storage sharing for social media content and the interest of different users. Through a novel ID code design that inherently reflects nodes' locations in an tree overlay, COOLS leverages stable storage users and yet inherently prioritizes living streaming flows, providing better scalability, robustness, and streaming quality.

Chapter 6

Load-Balanced Migration of Social Media to Cloud

Social media has been more and more popular, and has brought great challenges to the network engineering, particularly the huge demands of bandwidth and storage. The recently emerged cloud service sheds light on this dilemma. Towards the migration to cloud, partitioning the social media contents has drawn significant interests from the literature. Yet the existing works focus on preserving the social relationship only, while an important factor, user access pattern, is largely overlooked.

In this chapter, by examining a large collection of YouTube video data and Twitter user data, we first demonstrate that partitioning the network entirely based on social relationship would lead to unbalanced partitions in terms of access in Section 6.2. We further analyze the role of social relationship in the social media applications, and conclude that social relationship should be dynamically preserved. In Section 6.3, we formulate the problem as a constrained k-medoids clustering problem, and propose a novel solution in Section 6.4. In Section 6.5, we compare our solution with other state-of-the-art algorithms, and the results show that it significantly decreases the access deviation in each cloud server, and flexibly preserves the social relationship. We discuss the peer-to-peer scenario in Section 6.6. Finally, in Section 6.7, we conclude this chapter.

6.1 Introduction

Social media services have been dominating the Web 2.0 world in the recent years. Besides the most popular representatives such as YouTube, Facebook, and Twitter, many other social media services have emerged and are developing extremely fast. In general, it is difficult if not impossible to predict the impact and the development of any social media service in advance. The provision of resource is thus a great challenge, because any service with novel ideas, advanced techniques, and smart marketing strategies, is possible to grow to the similar scale to YouTube, Facebook, and Twitter. On the other hand, any service is possible to fail, losing revenue and even being shut down. The social media services have brought great challenges to the network engineering, and they face a great provisioning challenge due to the huge resource demands of bandwidth and storage.

The developers face a dilemma at the early stage of social media services. On one hand, to provision large enough resources at the beginning is costly and risky, and once the service is not popular as expected to gain enough revenue, the service will probably be failed. On the other hand, to start the service small usually comes with scalability issue. New features and increasing user base will cause the system suffer from the insufficient infrastructure, and thus further degrading the quality of service. We have already witnessed such applications as Friendster [44] and Myspace [73] failed due to the inscalability [29, 45], where both services were popular online social networking services before Facebook.

The recently emerged cloud service sheds light on this dilemma. A cloud service, such as Amazon EC2 [4], Microsoft Azure [111], Google App Engine [54], offers reliable, elastic and cost-effective resource provisioning, providing "pay-as-you-go" service that allows designers to start a service small and easy to scale big [5, 66]. Cloud users are charged based on the comupation, storage, and bandwidth. Some notable Internet services are currently cloud-based, e.g., Foursquare [43] (a location-based social networking services), Reddit [86] (a social news service), and Netflix [74] (an on-demand Internet streaming media).

Besides starting a service from cloud, a migration that moves the contents to cloud is essential for the benefit of the social media services' development and competition, since the these services will eventually face the scalability problem. In this thesis, we focus on the migration of current non-cloud contents to cloud, instead of designing a cloud-based application starting from scratch. Yet it is also an important issue to study in the future.

To move such social media service to cloud, one of the most important steps is to partition

the contents so as to assign them into a number of cloud servers. Different from traditional web contents that are isolated, social media contents have connections among each other, and thus the partition is non-trivial. There have been several existing works trying to solve this partitioning problem, such as SNAP [6] and SPAR [84]. Aiming at preserving the social relationship, they are quite effective tools to divide the social graph.

However, considering the cloud scenario, one important factor, user access pattern, is largely overlooked in the previous studies. If the contents (videos or users) are assigned into cloud servers entirely based on social relationship, a possible result is that some servers are holding many very popular contents but some are holding many unpopular ones, even if the load-balance in terms of the number is considered. By the evidence of our measurement study on a large collection of YouTube video data and Twitter user data, we demonstrate the existence of this phenomenon, which would cause great problem in cloud computing, especially from the aspect of network engineering. Specifically, in the client/server architecture system, some cloud servers with many popular contents will be accessed much more frequently than the other servers with unpopular contents, and this behavior will decrease the utilization of cloud computing, increasing unnecessary cost for the service providers [5]. Even worse, the cloud server may not handle the computation and transmission of the intensive data, while workloads of others are extremely low. The unbalanced partition will also lead to network traffic problems, which further degrades the quality of service.

On the other hand, partitioning the social media by taking user access pattern into account is not simply isolating the popular contents and evenly distributing the others. Taking some representatives of social media services as examples, we analyze the role of social relationship in these applications, and argue that it should be dynamically preserved. Therefore, our method should also take social relationship into account.

In this chapter, we formulate the problem as a constrained k-medoids clustering problem, and present a dissimilarity/similarity metric to facilitate the preservation of the social relationship. We propose a novel Weighted Partitioning Around Medoids (wPAM) algorithm to partition the social networked video repository, focusing on load-balance in terms of access. We evaluate our solution on YouTube data, comparing with the state-of-the-art algorithms. The results show that wPAM achieves extremely low deviation of load in terms of the popularity, and flexibly preserves the social relationship under different requirements.

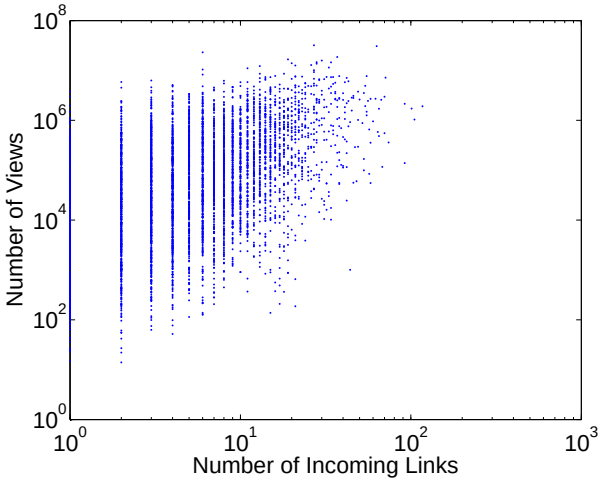


Figure 6.1: Popularity against incoming links

6.2 Motivation

In this section, by examining YouTube video data and Twitter user data, we show the correlation between the social relationship and the user access pattern. We argue that partitioning the social media entirely based on social relationship is not enough, as user access pattern should be taken into account. We also discuss the role of social relationship in various social media applications, and conclude that a dynamic preservation of social relationship is required.

6.2.1 Understanding User Access Pattern

User access pattern is yet to be considered while partitioning the social media. We show strong evidence from YouTube and Twitter that partitioning entirely based on social relationship will lead to unbalanced partitions in terms of popularity.

We have crawled YouTube videos and obtained a dataset containing over 40 thousand videos. The methodology of the data collection and the data format can be referred to our previous work [17, 25]. YouTube video graph is a directed graph, and the dataset only records the top-twenty outgoing related video IDs for each video, yet the incoming videos can be easily found within the dataset, and the number might be much greater than 20. We are particularly interested in the number of views and incoming links.

We have also collected Twitter data, which contains the public information of 1.47 million

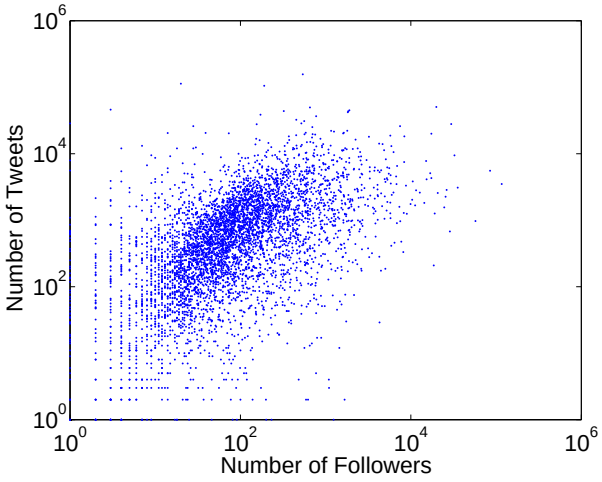


Figure 6.2: Number of tweets against number of followers in Twitter

Twitter users, such as the number of followers, the number of followed users, the number of tweets (statuses) posted, etc. Like YouTube, Twitter users' graph is also a directed graph. Unfortunately, our data only include the number of social relations, while the links of the followers and followed users are not available.

For the YouTube data, Figure 6.1 shows the scatter plot of the video's popularity against the number of incoming links. There is a clear trend that videos with more incoming links are more popular, because videos with more incoming links have more chances to be accessed through related videos. As we studied in Chapter 3, related video list is one of the most important video view sources. We also find that videos with few incoming links might also be very popular, because our measurement dataset does not cover the entire YouTube video repository. Nevertheless, videos with many social relations are mostly popular.

Figure 6.2 shows the correlation between the number of tweets and the number of followers (which is equivalent to the incoming links). The trend clearly appears again, and unlike YouTube, there are few user with small number of followers having posted great number of tweets. Nevertheless, both figures confirm that with more social relations, the contents are more likely to be popular; and from the other aspect, if the contents are popular, there will be more social relations.

We further examine the correlation between the video's popularity and the neighbours' popularity in YouTube. We calculate the mean of neighbours' views, and plot against the video views in Figure 6.3, which clearly shows the positive correlation. Figure 6.4 further

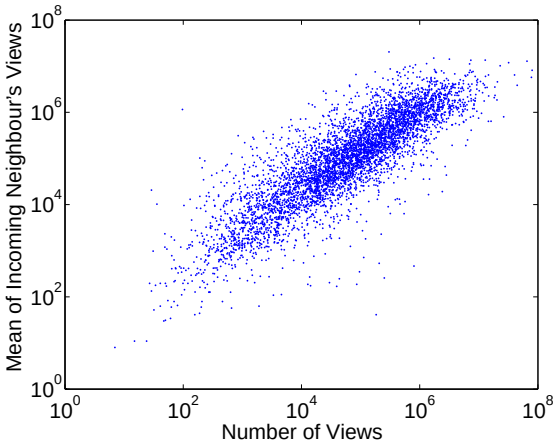


Figure 6.3: Mean of neighbour view against number of views

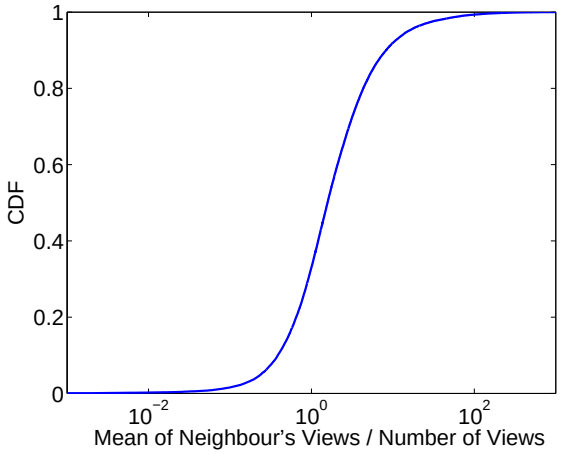


Figure 6.4: CDF of mean of neighbours' views over number of views

shows the CDF of the ratio of neighbours' popularity and the video's own popularity. Most of the videos have the comparable number of views as their neighbours' (ratio between 0.1 and 10). This characteristic indicates that if a video is popular, its neighbours are probably also popular, and vice versa.

In summary, a popular content's social neighbours are probably also popular, and they are likely to be clustered based on the social relationship only.

6.2.2 Social Relationship Is Not Enough

From the measurement above we know that partitioning the social media entirely based on social relationship would lead to unbalanced partitions, that is, some cloud servers have many very popular contents, but others have many very unpopular contents. This behaviour would cause great problems in the cloud as we discussed. Simply say, some cloud servers will have great or even overwhelming workload while some servers are nearly idle.

Figure 6.5 gives a simple example to make this phenomenon more clear: 8 nodes, each with a weight in terms of popularity as shown, constitute a social graph, which is going to be divided into two parts. As a result, the number of inter-connections is 2 of the left graph, but the standard deviation of the total weight in each part is as great as 348; while for the right graph that we take popularity into account, although the number of inter-connections increases to 4, the standard deviation is reduced to as small as 23, and thus the two parts will be accessed more evenly.

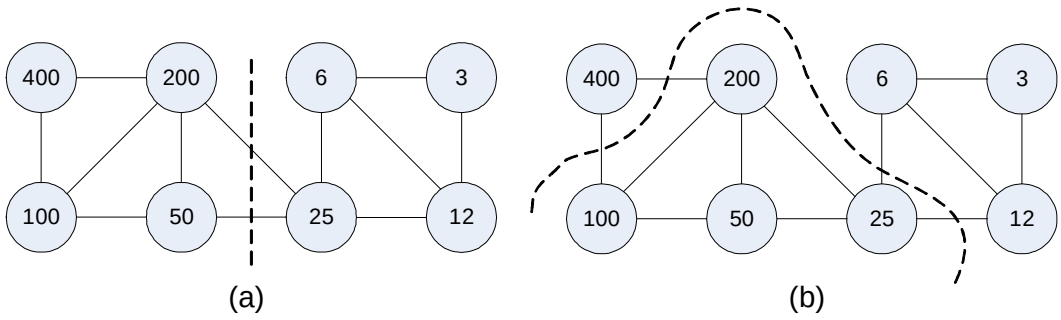


Figure 6.5: Example of different partitions based on (a) social relationship only, (b) both social relationship and popularity

6.2.3 Beyond Social Relationship

It is intuitive to preserve the social relationship when partitioning the social graph, because contents with social relationship are likely be accessed together or within a short period. If the two related contents are located in two different servers, it might increase the lookup time and the communication overhead. However, we raise our question: is social relationship *that* important? Towards answering this question, we analyze the role that social relationship plays in different types of social media applications, namely, video sharing service and social networking service.

Video Sharing Service

We take the representative YouTube as an example, and discuss videos' social relationship in particular. Each YouTube video has a list of related videos, which constitute a social graph. Two videos that have similar titles, tags, descriptions or from the same uploader are likely to be linked to each other. The related videos can also be considered as the recommendation from the YouTube system. As studied, the related video is likely to be requested after the user finishes watching the current video [21, 24, 127]. If two related videos are held by the same cloud server, the system can quickly locate the second one to achieve a smooth transition; if the two videos are held by two different cloud servers, this process will take longer, and thus delay will occur. Therefore from this perspective, preserving social relationship is beneficial.

However, the social relationship among videos is not so important that preserving it should be the top priority. As the measurement study [127] shows, about 30% of the overall

views are referred from the related recommendation, which is one of the most important view sources. This characteristic implies that the system's recommendation has a great influence on user behaviour of watching YouTube videos, and thus the system can easily adapt to the situation where related videos are in different cloud servers, e.g., by lowering the rank of the recommendation of the videos that are in different cloud servers. In fact, Zhou *et al.* [127] suggested utilizing YouTube recommendation to help increase the diversity of video views in aggregation, which encourages users to discover more videos of their interest rather than the popular videos only.

Furthermore, the pre-fetching mechanism had been proposed to reduce the startup delay for social media sharing [21, 24, 113]. In a pre-fetching mechanism, the next video is predicted and the prefix of the video is being fetched while the user is watching the first video. Considering a system utilizes such a mechanism, transmitting the two videos at the same time from the two respective cloud servers works better, because the mechanism takes advantage of the cloud computing to better utilize the network resource. Therefore from this perspective, **breaking the social relationship is beneficial**, which seems counter-intuitive but is the case under this circumstance.

Social Networking Service

We take Facebook as the first example. The social graph in Facebook is indirected, that is, if a is b's friend, b is also a's friend'; regarding Facebook, we are discussing about the users. When logs in, a Facebook user will receive a news feed that includes some of her/his friends' activities, e.g., updating on status, uploading photos, and sharing videos. Not all the friends' feed will be shown, as Facebook utilizes some algorithms to determine the ranking and provides to the user [40]. In this sense, retaining the social relationship, that is, keeping the close friends in the same cloud server, is somewhat important. However, even if they are in different cloud servers, new algorithm is easy to be re-designed without affecting the outcome much. In other word, retaining the social relationship in such social networking services as Facebook is good, but failing to do so does not affect the performance.

In Twitter, a user has friends to follow and is followed back, s/he also has celebrities and organizations to follow but is unlikely to be followed back, and s/he might be followed by someone not acquainted with. Unlike Facebook, a user's tweet is updated to the followers and s/he receives all the tweets posted by users that s/he follows. In this sense, the Twitter system does not selectively send tweets to a user, instead, the user retrieve all the available

tweets, and if the accounts followed by the user are in the same server, the lookup will be efficient. Since the tweets are text with small size compared to videos, sending from multiple servers will not help improve the transmission. Therefore, social relationship is important in such social networking services as Twitter.

In summary, the importance of preserving social relationship depends on different applications. For some applications, social relationship is not so significant, and thus it should not be the top priority. Specifically for video sharing service, some systems can easily adapt to the situation where related videos are held by different cloud servers, and if the systems utilize a pre-fetching mechanism, it is even better to break the social relationship. Therefore, we expect a solution that can partition the graph by taking user access pattern into account and can dynamically preserve the social relationship.

6.3 Problem Statement

In this section, we first formulate the problem statement, and then introduce a new distance metric facilitate the problem. We also discuss the details on weight constraint.

6.3.1 Formulation

Consider a social graph with N nodes $n_1, n_2, \ldots, n_N$, each node n_i has a weight w_i.

We try to partition the nodes into k clusters (cloud servers), $C_1, C_2, \ldots, C_k$. Each cluster C_j has a weight W_j, which is the summation of all the weights of the nodes in cluster C_j, that is,

$$W_j = \sum_{n_i \in C_j} w_i.$$

We suppose there is a representative node o_j in each cluster C_j. We denote $d(n_s, n_t)$ as a distance metric between the two node n_s and n_t.

The problem is to find a partition which minimizes

$$E = \sum_{j=1}^{k} \sum_{n_i \in C_j} d(n_i, o_j)$$

subject to

$$|W_{j_1} - W_{j_2}| < \Delta$$

for $j_1, j_2 = 1, 2, \ldots, k$, where Δ is a weight difference constraint.

The problem can be considered as a k-medoids clustering problem with constraint. The k-medoids problem is related to the k-means problem, except that the k-means calculates the mean in each cluster as the centre yet the k-medoids selects a representative node in each cluster as the centre. The k-medoids method is more robust than the k-means in the presence of noise and outliers, because a medoid is less influenced by outliers or other extreme values than a mean [56].

6.3.2 Node Distance – Dissimilarity/Similarity

In the previous studies, several metrics for measuring social networks were used, such as betweenness [75], conductance [71], modularity [6], and number of replicas [84]. Since k-medoids problem generally uses Euclidean distance as the metric, we initially considered using integral value of shortest path length between nodes as the metric. After further consideration, we found that the range of the integral path length is however too small. Our previous studies have shown that the shortest path length for YouTube dataset is about 8 [17, 25], and that individuals are separated by six degrees of social contact, known as "six degree of separation" [70]. Furthermore, to compute the path length between two nodes requires the knowledge of the entire social graph, which is very costly.

We thus introduce a pair of new metrics, *dissimilarity* and *similarity*. The metrics provide a much larger range of calculation than integral value of path length, and computing them only requires the knowledge of the adjacent nodes, which is much more efficient than requiring the information of the entire graph.

We define the similarity metric as follows: consider two nodes n_s and n_t, each having a set of adjacent nodes A_s and A_t, respectively. Let $A_s^* = A_s \cup n_s$ and $A_t^* = A_t \cup n_t$. The similarity is calculated as

$$\mathbf{sim}(n_s, n_t) = \frac{\left|A_s^* \bigcap A_t^*\right|}{\left|A_s^* \bigcup A_t^*\right|}.$$

Different from Jaccard similarity coefficient [58], where the similarity is calculated only based on the two adjacent node set, we include the node itself in the adjacent node set. The reason is that, if two target nodes are connected, they have direct social relation, while if they are not connected, even if they have identical adjacent node set, they do not have direct social relation. To examine the effect of the difference, we give an example in Figure 6.6, where both n_s and n_t have four adjacent nodes, yet they are not adjacent on the left while

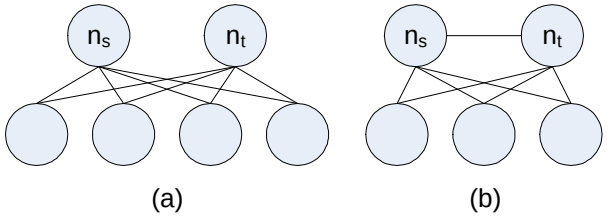


Figure 6.6: Similarity calculation

they are adjacent on the right. If we do not include the node itself in the adjacent node set, that is, the similarity is calculated by Jaccard similarity coefficient as $\frac{|A_s \bigcap A_t|}{|A_s \bigcup A_t|}$, the results are 1 and 0.6, respectively. However, the right graph is closer to our concept of "similar", because the two target nodes have social relationship. By our definition of similarity, the results are 0.67 and 1, respectively.

To cluster the similar nodes (such as for Twitter users), we require the smaller distance indicating closer nodes, and thus we use dissimilarity as the metric, which is defined as

$$\mathbf{dissim}(n_s, n_t) = 1 - \mathbf{sim}(n_s, n_t).$$

Therefore, the dissimilarity/similarity metric has the unique advantage: we can either use dissimilarity as the metric to preserve the social relationship, or use similarity as the metric to break the social relationship, that is, to cluster the dissimilar nodes.

6.3.3 Weight Constraint

The weight represents different metrics in various applications. For example, in such video sharing services as YouTube, the weight of a node is the number of views of the video, which reflects the possibility of being accessed by the users; in such social networking micro-blogging as Twitter, the weight of a node is the number of tweets posted, which reflects the activeness of the user; in such online social networking service as Facebook, there is no explicit metric representing as weight, but one can calculate an activeness of a user based on the activities.

We do not utilize standard deviation for the constraint, because the mathematical formula of the standard deviation (root square) will make the problem difficult if not entirely infeasible to solve; yet we will calculate the standard deviation for the evaluation results.

Since it is nearly impossible to decrease the standard deviation to zero, that is, all the clusters have the identical weight, it is better to use a threshold. In our problem statement, the difference between the total weight of any two clusters should be less than Δ.

Clearly, the smaller the threshold is, the tighter the constraint is, and the less the social relationship will be preserved (if using dissimilarity metric). Thus there exists a trade-off between social relationship preservation and load-balancing, and we will examine it in the evaluation by testing different value of Δ.

6.4 Solution

To solve the problem, we propose a novel algorithm in this section, and explain the modifications from the original classic PAM algorithm. We also discuss methods to improve its efficiency and scalability.

6.4.1 Weighted Partitioning Around Medoids: wPAM

Both k-medoids and k-means clustering problems have already been proven to be NP-hard, and a variety of heuristic algorithms have been proposed. *PAM* (Partitioning Around Medoids) was one of the first k-medoids algorithms introduced [61]. We have developed a Weighted PAM algorithm, *wPAM*, based on PAM, and Table 6.1 shows the pseudo-code.

Step (3) is the only modification from the original PAM, and it contains two major differences. First, in PAM, the distance is between the target node and the representative node in the cluster. We found that there is a great chance that the target node has no mutual neighbours with any representative nodes, and thus using dissimilarity/similarity metric would not get the nearest one. To address the problem, we not only calculate the distance between the target node and the representative node, but also the distance between the target node and the set of nodes that are in the cluster. Specifically, the target node is n_t, and the set of nodes in the cluster is A_s; we apply the distance calculation in Section 6.3, and further calculate the mean of the two distances.

Second, in PAM, each node is just assigned to the nearest cluster. Since we have weight constraint, when assigning the node, we need to ensure that the resulted cluster's weight will not exceed the maximum. In the perfect case, each cluster has a weight $\overline{W}$. While in the worst case, all clusters but one have weight $\overline{W} + \delta$, which is the maximum result from the algorithm, the remaining cluster thus has weight $\overline{W} - (k-1) \cdot \delta$. The difference between

Table 6.1: Pseudo-code of wPAM

Input: N nodes $n_1, n_2, \ldots, n_N$ with weight $w_1, w_2, \ldots, w_N$, cluster number k, weight constraint threshold δ.
Output: a set of k clusters satisfying the weight difference constraint: $\lvert W_{j_1} - W_{j_2} \rvert < k \cdot \delta$.
Method: 1: arbitrarily choose k nodes as the initial representative nodes (medoids); 2: **repeat** **3:** assign each remaining node with weight w, to the first nearest cluster with weight W_j, if satisfying $W_j + w < \overline{W} + \delta$, where $\overline{W}$ is the average weight of all the clusters; 4: randomly select a non-representative node, n'; 5: compute the total cost, $S = E' - E$, of swapping representative node, o_j, with n'; 6: if $S < 0$ then swap o_j with n' to form the new set of k representative nodes; 7: **until** no change;

the last cluster and the others is $k \cdot \delta = \Delta$, which is the weight difference threshold in the problem statement.

Although Step (5) is the same as the original algorithm, we exposit it in detail here. This step calculates the difference of E if a current representative node is replaced by a non-representative node,

$$E' - E = \sum_{n_i \in C_j} d(n_i, n') - \sum_{n_i \in C_j} d(n_i, o_j).$$

If this total cost is negative, then n_j is replaced or swapped with n' since the actual E would be reduced; if the total cost is positive, the current representative node, n_j, is considered acceptable, and nothing is changed in the iteration. From our experiment, we find that the number of iteration turns out to be under 5 in most cases.

6.4.2 Improving Efficiency

The complexity of iteration in PAM is $O(k(n-k)^2)$, where k is the number of clusters and n is the number of nodes. Therefore, it does not scale well for large dataset.

A sampling-based method, CLARA (Clustering LARge Applications), was introduced to deal with larger dataset [61]. The idea is taking a small portion of the data into consideration, choosing medoids from the sample using PAM. If the sample is selected fairly random, it should closely represent the original dataset, and the medoids chosen would likely be similar to those that would have been chosen from the whole dataset. The complexity of each iteration becomes $O(ks^2 + k(n-k))$, where s is the size of the sample.

CLARANS (Clustering Large Applications based upon RANdomized Search) was further proposed [76]. Unlike CLARA, CLARANS draws a sample with some randomness in each step of the search, and it has been experimentally shown to be more effective than both PAM and CLARA.

Both CLARA and CLARANS are based on PAM, and thus we can utilize either method based on our wPAM algorithm to solve the problem without any further modification. More details about CLARA and CLARANS can be referred to [61] and [76], respectively. To better understand the performance of wPAM, we do not utilize CLARA or CLARANS in the experiment in the next section.

6.5 Evaluation

We have implemented wPAM algorithm to evaluate its performance. In the experiment, we have examined different weight constraint, as we set Δ to be $0.1 \cdot \sum W$, $0.01 \cdot \sum W$ and $0.001 \cdot \sum W$. We have also examined different number of clusters (cloud servers), as we set k to be 4, 8, 16, 32, 64 and 128.

To compare, we have also implement three baseline methods. The first one is a random algorithm that assigns each video to a random cluster without considering either social relationship or weight constraint. The second one is the pLA algorithm in SNAP framework (Small-world Network Analysis and Partitioning) [6]. The pLA algorithm is a greedy aggregation algorithm that merges the nodes or clusters that increases the overall modularity score. A higher modularity score implies denser connections between the nodes within modules but sparser connections between nodes in different modules. The third method is SPAR algorithm [84]. In SPAR, if a node is assigned to a cluster, all the other clusters

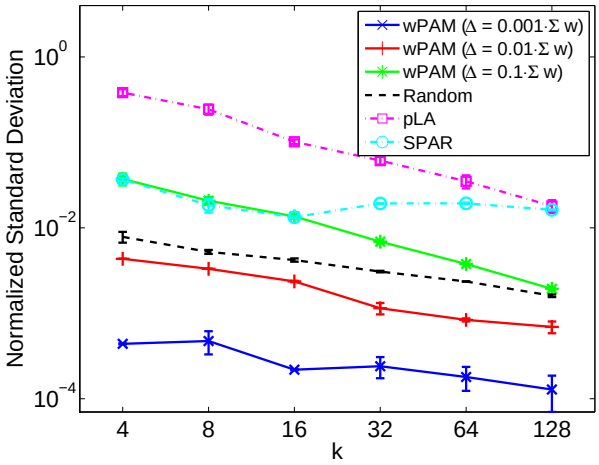


Figure 6.7: Comparison of normalized weight deviation

that have connections with the node will generate replicas of the node in the respective cluster; and therefore SPAR assigns each node to a cluster that needs to generate the fewest replicas. Since all the algorithms are randomized method, we run each algorithm five times and calculate the average results and their standard deviations.

Since our top priority is load-balance, we first look at the result of the standard deviation of each cluster's weight. The cluster's weight is the summation of all the videos' weight (popularity) in the cluster, and reflects the possibility of being accessed. The results against the number of clusters are shown in Figure 6.7. We normalize the result by dividing the total weight of all the videos in the dataset. Note that both axes are in logarithmic scale. pLA algorithm performs worst, because it tends to detect the popular community, and moreover, pLA does not consider load-balance in terms of the video number in each cluster, and thus the weight deviation is extremely high when the number of clusters is small. Random algorithm performs surprisingly well, because videos are randomly assigned and thus load-balance is partly achieved; however, random algorithm performs very bad on preserving social relationship as we will show next, and thus random algorithm still cannot be a solution if we take social relationship into account. SPAR algorithm performs worse than random algorithm. The reason is the same as that of pLA, yet SPAR strictly balances the number of videos in each cluster, and thus it performs better than pLA. When the weight difference constraint is loose and the number of clusters is small, wPAM algorithm performs similar to SPAR, and becomes better when the number of clusters increases. When

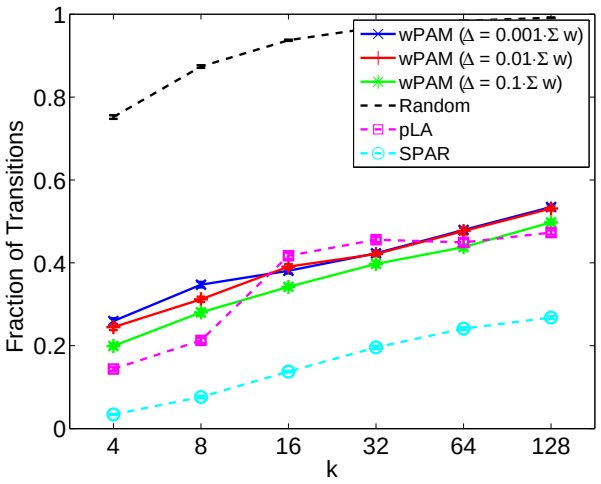


Figure 6.8: Comparison of fraction of transitions (preserving social relationship)

the constraint is tight, wPAM outperforms all the compared algorithms, as the green and blue solid lines shown in the figure, thanks to the strict constraint when clustering (step 3) in Table 6.1.

Supposing the system requires to preserve the social relationship as much as possible, we run wPAM using dissimilarity metric. To test how well the social relationship is preserved in the resulted partition, we generate 10 test cases, each contains $10,000$ YouTube video viewing transactions, which are used in our previous work [21, 24]. If two consecutive videos are held by two different cloud servers, we define it as one *transition*, and thus the less the number of transitions is, the better the social relationship is preserved. Because SPAR replicates all the linked nodes (called slave node) of a master node, the next video of a master video is always in the same server. Therefore, only if the next video of a slave video is in a different server, the number of transitions increases. We calculate the fraction of the transitions by dividing the total number $10,000$.

Figure 6.8 shows the results. Not surprisingly, random algorithm performs worst, because it does not take social relationship into account at all. On the other hand, SPAR performs best, since replicas are always in the same cloud server. Our wPAM performs similar to pLA, and with looser weight difference constraint, it performs better as expected, but the difference is not obvious. Although SPAR outperforms wPAM in terms of preserving social relationship, it requires much more space to store the replicas, about 4 to 12 times

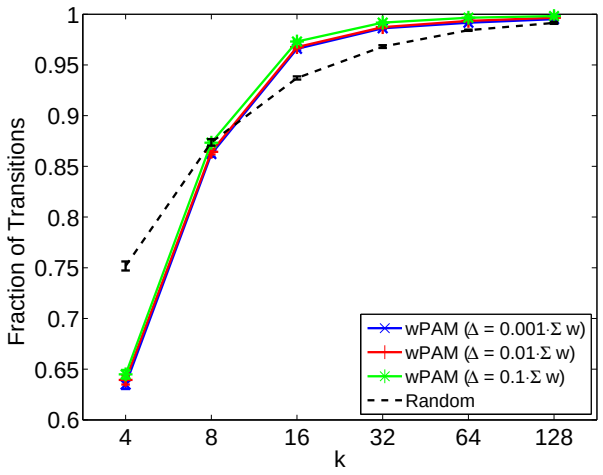


Figure 6.9: Comparison of fraction of transitions (breaking social relationship)

more than wPAM for different values of k. Nevertheless, wPAM achieves a rather good performance, even though our priority is load balance.

One advantage of wPAM is that we can break the social relationship to cluster the dissimilar videos, using similarity as the metric, which other algorithms are not capable to do. Figure 6.9 shows the result using the same test cases as above, and we compare it with the same random algorithm. As a result, when the number of clusters increases, wPAM outperforms random algorithm, and nearly achieves 100% transitions, that is, the next video is always in a different server. Looser weight difference constraints do not significantly improve the result. Therefore, considering the performance in terms of load-balance, we suggest using tight constraint when the system requires to break the social relationship.

6.6 Further Discussion

In the above study, especially for the video sharing service, the problem is in the scenario of client/server architecture, where popular contents are accessed more from the server perspective. Peer-to-peer technique has been widely used in data transmission, especially video streaming; for example, in Chapter 5, our COOLS system utilizes peer-to-peer to stream and share video. In peer-to-peer, videos are being shared among users in the same overlay network. It is known that peer-to-peer is scalable, that is, the more users participating in

the sharing, the greater downloading speed will be. If a content is unpopular, the overlay network is not large enough for effective downloading, the users can always turn to the server. Therefore, from the server perspective, in the peer-to-peer scenario, popular contents are less accessed.

The peer-to-peer scenario is much more complicated, as assigning weight to the nodes is a challenge. On one hand, the popularity of a video no longer indicates the probability it is accessed for all the videos from the server perspective, because popular video will be less accessed from the server; on the other hand, the weight is not simply reciprocal to the number of views, because for the unpopular videos, the probability of access is still determined by the number of views, since peer-to-peer overlays are too small to work for these unpopular videos.

To determine the weight of video in the peer-to-peer scenario, we suggest using empirical data. Specifically, we can conduct experiment based on our NetTube system [21, 24], and use server log to extract the number of accesses for each video from the server perspective. The same wPAM method can be utilized after assigning new weight to all the videos.

6.7 Conclusion

In this chapter, we took user access pattern into account in the problem of partitioning social media contents in the cloud scenario. By examining YouTube video data and Twitter user data, we demonstrated that partitioning social media content entirely based on social relationship leads to unbalanced access, which would cause great problem in cloud computing. We further concluded that a dynamic preservation of social relationship is preferred, by analyzing the role of social relationship in different social media services. We formulated the problem as a k-medoids clustering problem with constraint, and proposed a wPAM algorithm based on the classic PAM algorithm. The evaluation result show that wPAM significantly decreases the deviation of access in each cluster, and flexibly preserves the social relationship. We further discussed the partitioning problem in peer-to-peer scenario for migration of social media to cloud.

Chapter 7

Conclusion

Social media has emerged in the past decade. Among them, the representative of video sharing service, YouTube, and the representative of social networking service, Facebook and Twitter, have substantially changed the content distribution landscape, becoming one of the most important parts in people's everyday life. In this thesis, we have conducted measurement study to extend the existing research works to further understand social media, and proposed enhancements to augment social media service.

In this chapter, we first summarize this thesis, and then discuss the future directions.

7.1 Summary of the Thesis

As the most successful video sharing service, YouTube has started working with YouTube partners who upload high-quality premium videos and have a large audience. To help partners gain knowledge about their content accesses and audience behaviour, YouTube provides Insight data by showing simple scalars and charts. In this thesis, we effectively used Insight data to reveal a number of new features from the YouTube partners' view. We not only extended the existing research works on understanding the video sharing services, but also investigated the inherent relationship among various metrics that affect the popularity of the videos. In particular, we further examined the formation of the drop of the tail in views distribution; we proposed a simple yet effective method to identify viewing surges and further characterized them; we also examined visiting behaviour, daily pattern, user subscriptions and engagement activities; we revealed the breakdown of referral traffic of video views, and further investigated the impact from related videos and external sources

on the video views. Our findings on YouTube Insight data facilitate YouTube partners to adapt their content deployment and user engagement strategies, having great potentials for them to collaborate with YouTube to generate more views and subsequently increasing their revenues.

On the other hand, as video sharing services and social networking services become more and more integrated, video has become one of the most important types of object spreading among social network users. Therefore, we took an important step towards understanding the characteristics of video spreading in the social networks. Our measurement study was based on one week of 12.8 million video sharing and 115 million viewing event traces from Renren, the largest social networking service in China. We examined the user behaviour from diverse aspects, namely, initiating, watching, and sharing; we identified different users and further evaluated their activities. We also examined the temporal distribution during spreading as well as the typical propagation structures, revealing more details beyond stationary coverage. We extended the conventional epidemic SIR models to accommodate the diversity of the spreading, proposing an SI^2RP model to effectively capture the process of video spreading in the social networks. Our measurement on user behaviour and spreading structure facilitate the SI^2RP model. The large size and long duration of videos present significant challenges not only to the social networking service management, but also to the network traffic engineering. Our SI^2RP model can serve as a valuable tool for workload synthesis, traffic prediction, and resource provisioning.

Since video sharing social networking services become integrated, we have conducted a user questionnaire survey to directly understand user social behaviour and their video sharing preference. The survey results not only reveal an interesting coexistence of live streaming and storage sharing, but also show that the users are generally more interested in watching their friend's videos and collaboration is a rationale choice in this scenario. Motivated by the measurement study on video spreading and the user questionnaire survey, we proposed a novel coordinated live streaming and storage sharing system, COOLS. Utilizing a novel ID code design that inherently reflects nodes' locations in an tree overlay, COOLS leverages stable storage users and yet inherently prioritizes living streaming flows. We also enhanced the basic COOLS design to improve its robustness. It is known that the existing video sharing websites are facing critical server bottlenecks and the surges created by the social networking users would make the situation even worse. Compared with other state-of-the-art solutions, COOLS successfully takes advantage of the coexistence of live streaming

and storage sharing, providing better scalability, robustness, and streaming quality.

As the social media services have become more and more popular, they have brought great challenges to the network engineering, particularly the huge demands of bandwidth and storage. The cloud service sheds light on this dilemma, yet the existing works on partitioning social media contents focus on preserving the social relationship only, while an important factor, user access pattern, is largely overlooked. By examining YouTube video data and Twitter user data, we demonstrated that partitioning social media content entirely based on social relationship leads to unbalanced access, which will cause great problems in cloud computing. By analyzing the role of social relationship in different social media services, we concluded that a dynamic preservation of social relationship is preferred. We formulated the problem as a k-medoids clustering problem with constraint, and proposed a novel solution based on the classic PAM algorithm. We showed that our solution, wPAM, significantly decreases the deviation of access in each cluster, and flexibly preserves the social relationship. We also discussed the partitioning problem is peer-to-peer scenario for migration of social media to cloud.

7.2 Future Directions

There are still several directions worth studying on social media content distribution.

Further measurement on Insight data. The YouTube Insight data provide valuable information towards understanding video characteristics and user behaviour. Some issues are yet to be investigated. To name one, as keyword searching is one of the most important referral sources, understanding the impact of keyword has great value to the YouTube partners. The study can help partners to improve video rank in the search results, providing the videos better chance to be watched. Another work can be done to discover the factors that create viewing surges. In Chapter 3, we have proposed a method to identify viewing surges and characterized them, yet we did not further investigate what caused the viewing surge. This knowledge is also very valuable to YouTube partners for their marketing strategies.

Further understanding of video spreading. Our measurement was based on RenRen, the largest Facebook-like social networking website in China. Unfortunately, we do not have video watching and sharing traces from Facebook. Although RenRen is popular

and has 31 million active monthly user, it is not identical to Facebook, from culture to behaviour. Such measurement on worldwide social networking services like Facebook, Twitter, among others, will extend our study and provide more understanding on video spreading in social networks.

Further enhancement on COOLS. COOLS promotes the coexistence of the two types of users for video streaming, yet there are still several possible avenues to explore in this framework. To name one, since there are wired Internet users and wireless mobile users, their heterogeneity needs to be addressed in the overlay construction and maintenance. Specifically, Internet users have better network connection in terms of quality, bandwidth, and stability, and thus they should be considered to be located close to the source. Moreover, since mobile users are usually charged for data usage, such users are not always suitable for relaying data. These requirements call for sophisticated design of the architecture as well as the operations. In addition, COOLS is a system for live streaming and storage sharing, while coexistence of video on-demand and storage sharing is also important and worth investigating.

More on cloud. In this thesis, we investigate the migration issue that moves the social media contents into cloud. A cloud-based application design from the scratch also worth studying, especially as cloud become more and more popular and widely used. In addition, both migration to cloud and starting from cloud require sophisticated maintenance mechanism on adding, removing nodes (contents), and adding, removing links (relationships) between nodes.

Appendix A

Database Schema and SQL Queries

A.1 Database Schema

The database schema is shown in Figure A.1. Table `entry` records all the videos and channels information, in which `entryid` is the primary key; Table `views_`*`(channel)`* and Table `views_`*`(channel)`*`_channel` record views data of videos and channels, respectively, and the primary keys are `(entryid, date)` and `date`, respectively; Table `referrers_`*`(channel)`* records referrers data of videos, in which `(entryid, date, source, detail)` is the primary key. Note that, *(channel)* is replaced with the corresponding channel name while using.

A.2 SQL Queries

The following lists all the SQL queries that are used to obtain partial data or final result.

- Get video lifetime statistics and age:

```
SELECT SUM(views),SUM(likes),SUM(dislikes),SUM(favourites),
SUM(comments),DATEDIFF(MAX(date),MIN(date))
FROM views_(channel)
GROUP BY entryid;
```

- Get video views for one month (February 2012):

```
SELECT SUM(views)
FROM views_(channel)
```

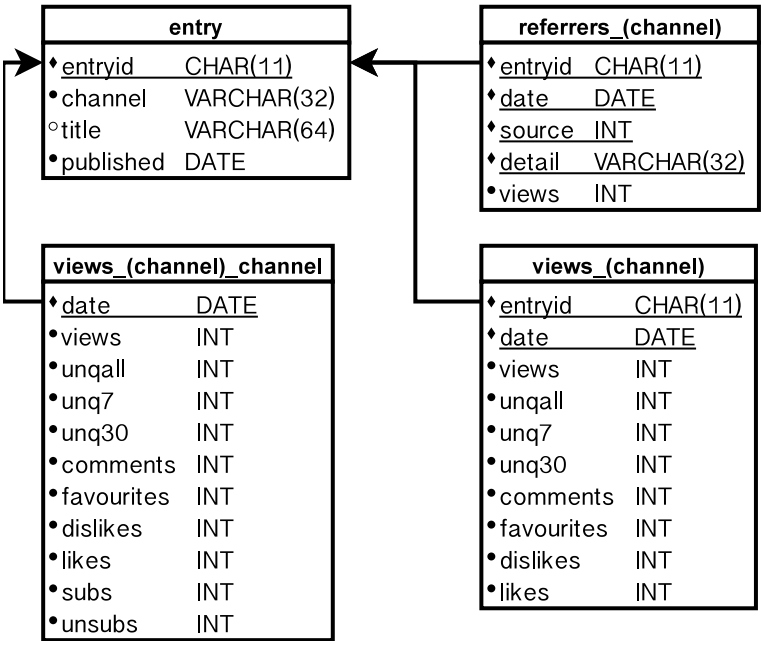


Figure A.1: Database schema

```
WHERE date>="2012-02-01" AND date<"2012-03-01"
GROUP BY entryid;
```

- Get views per unique visit:

```
SELECT AVG(viewpervisit)
FROM (SELECT entryid,date,views/unqall AS viewpervisit
FROM views_(channel)) tb
GROUP BY entryid;
```

- Get daily pattern in terms of video views:

```
SELECT
SUM(CASE WHEN DAYOFWEEK(date)=2 THEN views ELSE 0 END)/SUM(views)
SUM(CASE WHEN DAYOFWEEK(date)=3 THEN views ELSE 0 END)/SUM(views)
SUM(CASE WHEN DAYOFWEEK(date)=4 THEN views ELSE 0 END)/SUM(views)
SUM(CASE WHEN DAYOFWEEK(date)=5 THEN views ELSE 0 END)/SUM(views)
SUM(CASE WHEN DAYOFWEEK(date)=6 THEN views ELSE 0 END)/SUM(views)
SUM(CASE WHEN DAYOFWEEK(date)=7 THEN views ELSE 0 END)/SUM(views)
```

```
SUM(CASE WHEN DAYOFWEEK(date)=1 THEN views ELSE 0 END)/SUM(views)
FROM views_(channel);
```

- Get unique visitors statistics for each video:

```
SELECT tba.entryid,DATEDIFF(date,published),unqall,unq7,unq30
FROM views_(channel) tba INNER JOIN (
SELECT entryid, MIN(date) AS published
FROM views_(channel) GROUP BY entryid) tbb
ON tba.entryid = tbb.entryid;
```

- Get unique visitors statistics for each channel:

```
SELECT "(channel)",DATEDIFF(date,MIN(date)) AS age,
unqall,unq7,unq30
FROM views_(channel)_channel;
```

- Get views, subscription, and unsubscription along time:

```
SELECT SUM(views),SUM(subs),SUM(unsubs)
FROM views_(channel)_channel
GROUP BY date;
```

- Get breakdown of referrers:

```
SELECT SUM(CASE WHEN sourceid IN (9) THEN views ELSE 0 END),
SUM(CASE WHEN sourceid IN (4,13) THEN views ELSE 0 END),
SUM(CASE WHEN sourceid IN (1,2,8,10,11,12) THEN views ELSE 0 END),
SUM(CASE WHEN sourceid IN (3,5) THEN views ELSE 0 END),
SUM(CASE WHEN sourceid IN (6,7) THEN views ELSE 0 END)
FROM referrers_(channel);
```

- Get number of related videos that are in the same channel and network:

```
SELECT SUM(CASE WHEN detail IN (
SELECT entryid FROM entry WHERE channel = "(channel)")
THEN views ELSE 0 END),
SUM(CASE WHEN detail IN (SELECT entryid FROM entry)
THEN views ELSE 0 END),
SUM(views)
FROM (SELECT detail, SUM(views) AS views
```

```
FROM referrers_(channel)
WHERE sourceid = 9 GROUP BY detail) tb;
```

- Get top-5 external referrers:

```
SELECT detail,SUM(views)/(
SELECT SUM(views) FROM referrers_(channel) WHERE sourceid IN (3,5))
AS viewspct
FROM referrers_(channel)
WHERE sourceid IN (3,5)
GROUP BY detail ORDER BY viewspct DESC LIMIT 0, 5;
```

- Get percentage of external source views within 30 days:

```
SELECT AVG(viewsext/viewsall)
FROM (SELECT tba.entryid, DATEDIFF(dater,tbb.published) AS age,
(SUM(CASE WHEN sourceid IN (3,5) THEN views ELSE 0 END)) AS viewsext,
SUM(views) AS viewsall
FROM referrers_(channel) tba JOIN (
SELECT entryid, published
FROM entry
WHERE channel = "(channel)") tbb
ON tba.entryid = tbb.entryid
WHERE DATEDIFF(date,tbb.published) >=0 AND
DATEDIFF(date,tbb.published) < 30)
GROUP BY tba.entryid,age) tb
GROUP BY age ORDER BY age.
```

Appendix B

User Questionnaire Survey

1. In most cases, do you satisfy with the startup delay (delay from selecting the video to starting playing the video)?
   A. Yes; B. No.

2. In most cases, do you satisfy with the playback continuity (smoothness of playing the video)?
   A. Yes; B. No.

3. Which one best describes you in most cases?
   A. select the video, wait until start, do not leave; B. select the video, pause and leave to do something else, then return after a while and play.

4. Do you think the playback quality (startup delay and playback continuity) affects your behaviour in Question3?
   A. Yes; B. No.

5. If you leave, when will you return in most cases? If you do not leave, check n/a.
   A. n/a; B. 1/4 of the video that has been downloaded (independent on the video length); C. 1/2 ...; D. 3/4 ...; E. entire ...; F. <1 min (depending on the video length); G. 1-5 min; H. 5-10 min; I. 10-30 min; J. >30 min.

6. In most cases, while watching the video, do you jump to other playback positions?
   A. every time; B. most time; C. seldom; D. never.

7. In most cases, how long will you watch?
   A. 1/4 of the video; B. 1/2; C. 3/4; D. entire video.

8. What are the factors affecting your aforementioned behaviours?
   A. only playback quality; B. only video content; C. both, quality > content; D. both, quality < content.

9. If your friend uploads a video, will you watch the entire video?
   A. 100%; B. probably; C. likely; D. maybe; E. don't care at all.

10. If someone you are following (like in Twitter) uploads a video, will you watch the entire video?
    A. 100%; B. probably; C. likely; D. maybe; E. don't care at all.

11. When you are watching online videos, do you want to upload what you are watching to others?
    A. Yes, no matter who; B. Yes, if others are friends; C. No.

12. Have you ever watched online video through mobile devices (e.g., cellphones)?
    A. Yes; B. No.

13. How long do you spend on watching online video per day?
    A. <5 min; B. 5-10 min; C. 10-30 min; D. 30-60 min; E. 1-2 hr; F. 2-4 hr; G. >4 hr.

14. Your location
    A. North America; B. Europe; C. Asia; D. South America/Africa/Oceania.

15. Your age
    A. <19; B. 19-30; C. 31-50; D. >50.

16. Your network connection
    A. DSL; B. Cable Modem; C. Ethernet; D. Fibre; E. WiFi; F. 3G wireless; G. unknown.

Bibliography

[1] Adobe Flex. http://www.adobe.com/products/flex.html. (available as of June 8th, 2012).

[2] Chloe Albanesius. Inauguration: Twitter Reports 5 Times Normal Tweets Per Second. http://www.appscout.com/2009/01/inauguration_twitter_reports_5.php, 2009. (available as of June 8th, 2012).

[3] Alexa. http://www.alexa.com.

[4] Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2.

[5] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the Clouds: A Berkeley View of Cloud Computing. Technical report, University of California at Berkeley, 2009.

[6] David A. Bader and Kamesh Madduri. SNAP, Small-world Network Analysis and Partitioning: An Open-source Parallel Graph Framework for the Exploration of Large-Scale Networks. In *Proceedings of IEEE International Symposium on Parallel and Distributed Processing (IPDPS '08)*, pages 1–12, April 2008.

[7] Fabrício Benevenuto, Tiago Rodrigues, Meeyoung Cha, and Virgílio Almeida. Characterizing User Behavior in Online Social Networks. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement (IMC '09)*, pages 49–62, November 2009.

[8] BitTorrent Launches Ad Supported Streaming. http://torrentfreak.com/bittorrent-launches-ad-supported-streaming-071218, 2007. (available as of June 8th, 2012).

[9] Burton Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Communications of the ACM*, 13(7):422–426, July 1970.

[10] Peter Bloomfield. *Fourier Analysis of Time Series: An Introduction.* John Wiley & Sons, 2004.

[11] BroadbandTV Corp. http://broadbandtvcorp.com.

[12] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. Limiting the Spread of Misinformation in Social Networks. In *Proceedings of the 20th International Conference on World Wide Web (WWW '11)*, pages 665–674, April 2011.

[13] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC '07)*, pages 1–14, October 2007.

[14] Meeyoung Cha, Alan Mislove, and Krishna P. Gummadi. A Measurement-driven Analysis of Information Propagation in the Flickr Social Network. In *Proceedings of the 20th International Conference on World Wide Web (WWW '09)*, pages 721–730, April 2009.

[15] Xu Cheng. On the Characteristics and Enhancement of Internet Short Video Sharing. Master's thesis, Simon Fraser University, April 2008.

[16] Xu Cheng, Cameron Dale, and Jiangchuan Liu. Characteristics and Potentials of YouTube: A Measurement Study. In Eli M. Noam and Lorenzo Maria Pupillo, editors, *Peer-to-Peer Video: The Economics, Policy, and Culture of Today's New Mass Medium*, pages 205–217. Springer, September 2008.

[17] Xu Cheng, Cameron Dale, and Jiangchuan Liu. Statistics and Social Network of YouTube Videos. In *Proceedings of 16th International Workshop on Quality of Service (IWQoS '08)*, pages 229–238, June 2008.

[18] Xu Cheng, Mehrdad Fatourechi, and Jiangchuan Liu. *Insight Data of YouTube: From a Partner's View*. BroadbandTV Corp., 2012.

[19] Xu Cheng, Kunfeng Lai, Dan Wang, and Jiangchuan Liu. UGC Video Sharing: Measurement and Analysis. In Chang Wen Chen, Zhu Li, and Shiguo Lian, editors, *Intelligent Multimedia Communication: Techniques and Applications*, pages 367–402. Springer, August 2010.

[20] Xu Cheng, Haitao Li, and Jiangchuan Liu. Video Sharing Propagation in Social Networks: Measurement, Modeling, and Analysis. Under revision for journal publication, 2012.

[21] Xu Cheng and Jiangchuan Liu. NetTube: Exploring Social Networks for Peer-to-Peer Short Video Sharing. In *Proceedings of IEEE INFOCOM*, pages 1152–1160, April 2009.

[22] Xu Cheng and Jiangchuan Liu. Tweeting Videos: Coordinate Live Streaming and Storage Sharing. In *Proceedings of the 20th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '10)*, pages 15–20, June 2010.

[23] Xu Cheng and Jiangchuan Liu. Load-Balanced Migration of Social Media to Content Clouds. In *Proceedings of the 20th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '11)*, pages 51–56, June 2011.

[24] Xu Cheng and Jiangchuan Liu. Exploring Interest Correlation for Peer-to-Peer Socialized Video Sharing. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 8(1):5:1–5:20, January 2012.

[25] Xu Cheng, Jiangchuan Liu, and Cameron Dale. Understanding the Characteristics of Internet Short Video Sharing: A YouTube-based Measurement Study. To appear in IEEE Transactions on Multimedia, 2012.

[26] Xu Cheng, Jiangchuan Liu, and Haiyang Wang. Accelerating YouTube with Video Correlation. In *Proceedings of the 1st SIGMM Workshop on Social media (WSM '09)*, pages 49–56, October 2009.

[27] Xu Cheng, Jiangchuan Liu, Haiyang Wang, and Chonggang Wang. Coordinate Live Streaming and Storage Sharing for Social Media Content Distribution. To appear in IEEE Transactions on Multimedia, 2012.

[28] Xu Cheng, Feng Wang, Jiangchuan Liu, and Ke Xu. Collaborative Delay-Aware Scheduling in Peer-to-Peer UGC Video Sharing. In *Proceedings of the 20th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '10)*, pages 105–110, June 2010.

[29] Dawn C. Chmielewski and David Sarno. How MySpace fell off the pace. http://articles.latimes.com/2009/jun/17/business/fi-ct-myspace17, 2009. (available as of June 8th, 2012).

[30] Coefficient of determination. http://en.wikipedia.org/wiki/Coefficient_of_determination.

[31] Josh Constine. Facebook's Amended S-1: 901 Million Users, 500M Mobile, Paid $300M Cash + 23M Shares For Instagram. http://techcrunch.com/2012/04/23/facebooks-amended-s-1-500-million-mobile-users-paid-300m-cash-23-million-shares-for-instagram, 2012. (available as of June 8th, 2012).

[32] Douglas Crockford. "The application/json Media Type for JavaScript Object Notation (JSON). http://tools.ietf.org/html/rfc4627, 2006. (available as of June 8th, 2012).

[33] Dailymotion. http://www.dailymotion.com.

[34] Daryl J. Daley, Joe Gani, and Joseph Mark Gani. *Epidemic Modelling: An Introduction*. Cambridge Studies in Mathematical Biology. Cambridge University Press, 2001.

[35] Delicious. http://delicious.com.

[36] Kirill Dyagilev, Shie Mannor, and Elad Yom-Tov. Generative Models for Rapid Information Propagation. In *Proceedings of the First Workshop on Social Media Analytics (SOMA '10)*, pages 35–43, July 2010.

[37] Facebook. http://www.facebook.com.

[38] Facebook. Form S-1 REGISTRATION STATEMENT. http://www.sec.gov/Archives/edgar/data/1326801/000119312512034517/d287954ds1.htm, 2012. (available as of June 8th, 2012).

[39] Facebook Newsroom - Fact Sheet. http://newsroom.fb.com/content/default.aspx?NewsAreaId=22. (available as of June 8th, 2012).

[40] Facebook Help Center. http://www.facebook.com/help.

[41] Flavio Figueiredo, Fabrício Benevenuto, and Jussara M. Almeida. The Tube over Time: Characterizing Popularity Growth of YouTube Videos. In *Proceedings of the fourth ACM International Conference on Web Search and Data Mining (WSDM '11)*, pages 745–754, February 2011.

[42] Alessandro Finamore, Marco Mellia, Maurizio M. Munafò, Ruben Torres, and Sanjay G. Rao. YouTube Everywhere: Impact of Device and Infrastructure Synergies on User Experience. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement (IMC '11)*, pages 345–360, November 2011.

[43] Foursquare. http://foursquare.com.

[44] Friendster. http://www.friendster.com.

[45] Friendster Lost Lead Because Of A Failure To Scale. http://highscalability.com/blog/2007/11/13/friendster-lost-lead-because-of-a-failure-to-scale.html, 2007. (available as of June 8th, 2012).

[46] Gamma Distribution. http://en.wikipedia.org/wiki/Gamma_distribution.

[47] Ayalvadi J. Ganesh, Laurent Massoulié, and Don Towsley. The Effect of Network Topology on the Spread of Epidemics. In *Proceedings of IEEE INFOCOM*, pages 1455–1466, March 2005.

[48] Jesse James Garrett. Ajax: A New Approach to Web Applications. http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications, 2005. (available as of June 8th, 2012).

[49] Magdalena Georgieva. Americans Spend Nearly 3.5 Hours Per Week Watching Online Video [Data]. http://blog.hubspot.com/blog/tabid/6307/bid/11888/Americans-Spend-Nearly-3-5-Hours-Per-Week-Watching-Online-Video-Data.aspx, 2011. (available as of June 8th, 2012).

[50] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. YouTube Traffic Characterization: A View From the Edge. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC '07)*, pages 15–28, October 2007.

[51] Josh Goldman. YouTube Serves 100m Videos Each Day. http://techcrunch.com/2006/07/17/youtube-serves-100m-videos-each-day, 2006. (available as of June 8th, 2012).

[52] Google To Acquire YouTube for $1.65 Billion in Stock. http://www.google.com/press/pressrel/google_youtube.html, 2006. (available as of June 8th, 2012).

[53] Google+. http://plus.google.com.

[54] Google App Engine. http://code.google.com/appengine.

[55] Gonca Gürsun, Mark Crovella, and Ibrahim Matta. Describing and Forecasting Video Access Patterns. In *Proceedings IEEE INFOCOM 2011*, pages 16–20, April 2011.

[56] Jiawei Han and Micheline Kamber. *Data Mining Concepts and Techniques (2nd Edition)*. Morgan Kaufmann, 2006.

[57] Yan Huang, Tom Z.J. Fu, Dah-Ming Chiu, John C.S. Lui, and Cheng Huang. Challenges, Design and Analysis of a Large-scale P2P-VoD System. In *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, pages 375–388, August 2008.

[58] Jaccard Index. http://en.wikipedia.org/wiki/Jaccard_index.

[59] Jing Jiang, Christo Wilson, Xiao Wang, Peng Huang, Wenpeng Sha, Yafei Dai, and Ben Y. Zhao. Understanding Latent Interactions in Online Social Networks. In *Proceedings of the 10th annual conference on Internet measurement (IMC '10)*, pages 369–382, November 2010.

[60] Andreas M. Kaplan and Michael Haenlein. Users of the World, Unite! The Challenges and Opportunities of Social Media. *Business Horizons*, 53(1):59–68, January–Febuary 2010.

[61] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 1990.

[62] William O. Kermack and Anderson G. McKendrick. A Contribution to the Mathematical Theory of Epidemics. *Proceedings of the Royal Society of London. Series A*, 115(772):700–721, 1927.

[63] Jan H. Kietzmann, Kristopher Hermkens, Ian P. McCarthy, and Bruno S. Silvestre. Social Media? Get Serious! Understanding the Functional Building Blocks of Social Media. *Business Horizons (Special Issue: Social Media)*, 54(3):241–251, May-June 2011.

[64] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a Social Network or a News Media? In *Proceedings of the 19th International World Wide Web Conference (WWW '10)*, pages 591–600, April 2010.

[65] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the Dynamics of the News Cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 497–506, June 2009.

[66] Ang Li, Xiaowei Yang, Srikanth Kandula, and Ming Zhang. CloudCmp: Comparing Public Cloud Providers. In *Proceedings of the 10th annual Conference on Internet Measurement (IMC '10)*, pages 1–14, November 2010.

[67] Jiangchuan Liu, Sanjay G. Rao, Bo Li, and Hui Zhang. Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast. *Proceedings of the IEEE*, 96(1):11–24, January 2008.

[68] Zonghua Liu, Ying-Cheng Lai, and Nong Ye. Propagation and Immunization of Infection on General Networks with Both Homogeneous and Heterogeneous Components. *Physical Review E*, 67(1):031911, March 2003.

[69] Metacafe. http://www.metacafe.com.

[70] Stanley Milgram. The Small World Problem. *Psychology Today*, 2(1):60–67, 1967.

[71] Nina Mishra, Robert Schreiber, Isabelle Stanton, and Robert Tarjan. Clustering Social Networks. In *Algorithms and Models for the Web-Graph*, volume 4863 of *Lecture Notes in Computer Science*, pages 56–67. Springer Berlin/Heidelber, November–December 2006.

[72] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Dreschel, and Bobby Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC '07)*, pages 29–42, October 2007.

[73] Myspace. http://myspace.com.

[74] Netflix. http://www.netflix.com.

[75] Mark E.J. Newman and Michelle Girvan. Finding and Evaluating Community Structure in Networks. *Physical Review E*, 69(2):026113, February 2004.

[76] Raymond T. Ng and Jiawei Han. Efficient and Effective Clustering Methods for Spatial Data Mining. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94)*, pages 144–155, September 1994.

[77] Official Google Blog. Shipping the Google in Google+. http://googleblog.blogspot.com/2011/11/shipping-google-in-google.html, 2011. (available as of June 8th, 2012).

[78] Megan O'Neill. Cisco Predicts That 90% Of All Internet Traffic Will Be Video In The Next Three Years. http://socialtimes.com/cisco-predicts-that-90-of-all-internet-traffic-will-be-video-in-the-next-three-years_b82819, 2011. (available as of June 8th, 2012).

[79] Tim O'Reilly. What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. http://oreilly.com/web2/archive/what-is-web-20.html, 2005. (available as of June 8th, 2012).

[80] Pajek. http://vlado.fmf.uni-lj.si/pub/networks/pajek.

[81] Pareto Distribution. http://en.wikipedia.org/wiki/Pareto_distribution.

[82] Pearson product-moment correlation coefficient. http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient.

[83] PlanetLab. www.planet-lab.org.

[84] Josep M. Pujol, Vijay Erramilli, Georgos Siganos, Xiaoyuan Yang, Nikos Laoutaris, Parminder Chhabra, and Pablo Rodriguez. The Little Engine(s) That Could: Scaling Online Social Networks. In *Proceedings of the ACM SIGCOMM 2010 Conference*, pages 375–386, August 2010.

[85] Quora. http://www.quora.com.

[86] Reddit. http://www.reddit.com.

[87] Renren. http://www.renren.com.

[88] Renren. http://en.wikipedia.org/wiki/Renren.

[89] Tiago Rodrigues, Fabrício Benevenuto, Meeyoung Cha, Krishna P. Gummadi, and Virgílio Almeida. On Word-of-Mouth Based Discovery of the Web. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement (IMC '11)*, pages 381–396, November 2011.

[90] Matt Rosoff. Twitter Just Had Its CNN Moment. http://www.businessinsider.com/twitter-just-had-its-cnn-moment-2011-5, 2011. (available as of June 8th, 2012).

[91] Fabian Schneider, Anja Feldmann, Balachander Krishnamurthy, and Walter Willinger. Understanding Online Social Network Usage from a Network Perspective. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement (IMC '09)*, pages 35–48, November 2009.

[92] Maggie Shiels. Twitter Co-founder Jack Dorsey Rejoins Company. http://www.bbc.co.uk/news/business-12889048, 2011. (available as of June 8th, 2012).

[93] MG Siegler. News Faster than News Outlets: Why the Internet (and Twitter) Wins. http://seekingalpha.com/article/175573-news-faster-than-news-outlets-why-the-internet-and-twitter-wins, 2009. (available as of June 8th, 2012).

[94] Social Media Report: Q3 2011. http://blog.nielsen.com/nielsenwire/social, 2011. (available as of June 8th, 2012).

[95] Social network. http://en.wikipedia.org/wiki/Social_network.

[96] Brian Stelter. Google Is Said to Weigh Investing in Machinima, a Creator for YouTube. http://mediadecoder.blogs.nytimes.com/2012/05/07/google-is-said-to-weigh-investing-in-machinima-a-creator-for-youtube/, 2012. (available as of June 8th, 2012).

[97] Travis Steven Stiles. Web 2.0, The Meta-Meme or The Present-Future of the Web. http://www.tstiles.com/dms/web20/index.html, 2005. (available as of June 8th, 2012).

[98] Shaojie Tang, Jing Yuan, Xufei Mao, Xiang-Yang Li, Wei Chen, and Guojun Dai. Relationship Classification in Large Scale Online Social Networks and Its Impact on Information Propagation. In *Proceedings of IEEE INFOCOM*, pages 2291–2299, April 2011.

[99] Joseph Tartakoff. Twitter Search Fails Under Thursday's Celebrity News Rush. http://paidcontent.org/article/419-twitter-search-fails-under-thursdays-celebrity-news-rush, 2009. (available as of June 8th, 2012).

[100] The Smartphone Will Dominate Internet Access by 2020. http://www.cellular-news.com/story/35160.php. (available as of June 8th, 2012).

[101] Top 10 Internet Memes. http://www.squidoo.com/top-10-internet-memes. (available as of June 8th, 2012).

[102] Tudou. www.tudou.com.

[103] Twitter. http://twitter.com.

[104] Twitter Blog. Your world, more connected. http://blog.twitter.com/2011/08/your-world-more-connected.html, 2011. (available as of June 8th, 2012).

[105] Vidhyashankar Venkataraman, Kaouru Yoshida, and Paul Francis. Chunkyspread: Heterogeneous Unstructured Tree-Based Peer-to-Peer Multicast. In *Proceedings of the 2006 14th IEEE International Conference on Network Protocols (ICNP '06)*, pages 2–11, November 2006.

[106] Vimeo. http://vimeo.com.

[107] Dashun Wang, Zhen Wen, Hanghang Tong, Ching-Yung Lin, Chaoming Song, and Albert-Laszlo Barabasi. Information Spreading in Context. In *Proceedings of the 20th International Conference on World Wide Web (WWW '11)*, pages 735–744, April 2011.

[108] Feng Wang, Jiangchuan Liu, and Yongqiang Xiong. On Node Stability and Organization in Peer-to-Peer Video Streaming Systems. *IEEE Systems Journal*, 5(4):440–450, December 2011.

[109] Weibull Distribution. http://en.wikipedia.org/wiki/Weibull_distribution.

[110] Wikipedia. http://www.wikipedia.org/.

[111] Windows Azure. http://www.windowsazure.com.

[112] Chad Wittman. Comments 4x More Valuable Than Likes. http://edgerankchecker.com/blog/2011/11/comments-4x-more-valuable-than-likes, 2011. (available as of June 8th, 2012).

[113] Ke Xu, Haitao Li, Jiangchuan Liu, Wei Zhu, and Wenyu Wang. PPVA: A Universal and Transparent Peer-to-Peer Accelerator for Interactive Online Video Sharing. In *Proceedings of 18th International Workshop on Quality of Service (IWQoS '10)*, pages 1–9, June 2010.

[114] Youku. http://www.youku.com.

[115] Youtube. http://www.youtube.com.

[116] YouTube Blog. YouTube Elevates Most Popular Users to Partners. http://youtube-global.blogspot.com/2007/05/youtube-elevates-most-popular-users-to.html, 2007. (available as of June 8th, 2012).

[117] YouTube Blog. YouTube Reveals Video Analytics Tool for All Users. http://youtube-global.blogspot.com/2008/03/youtube-reveals-video-analytics-tool.html, 2008. (available as of June 8th, 2012).

[118] YouTube Blog. Y,000,000,000uTube. http://youtube-global.blogspot.com/2009/10/y000000000utube.html, 2009. (available as of June 8th, 2012).

[119] YouTube Blog. At Five Years, Two Billion Views Per Day and Counting. http://youtube-global.blogspot.com/2010/05/at-five-years-two-billion-views-per-day.html, 2010. (available as of June 8th, 2012).

[120] YouTube Blog. Thanks, YouTube Community, for Two BIG Gifts on Our Sixth Birthday! http://youtube-global.blogspot.com/2011/05/thanks-youtube-community-for-two-big.html, 2011. (available as of June 8th, 2012).

[121] YouTube Blog. Holy Nyans! 60 Hours Per Minute and 4 Billion Views a Day on YouTube. http://youtube-global.blogspot.com/2012/01/holy-nyans-60-hours-per-minute-and-4.html, 2012. (available as of June 8th, 2012).

[122] YouTube Creator Playbook. http://www.youtube.com/creators/playbook.html.

[123] YouTube - Google Developers. https://developers.google.com/youtube.

[124] YouTube - kidrauhl's Channel. http://www.youtube.com/user/kidrauhl.

[125] YouTube Press - Statistics. http://www.youtube.com/t/press_statistics.

[126] Xinyan Zhang, Jiangchuan Liu, Bo Li, and Tak-Shing Peter Yum. CoolStreaming/DONet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming. In *Proceedings of IEEE INFOCOM*, volume 3, pages 2102 – 2111, March 2005.

[127] Renjie Zhou, Samamon Khemmarat, and Lixin Gao. The Impact of YouTube Recommendation System on Video Views. In *Proceedings of the 10th annual conference on Internet measurement (IMC '10)*, pages 404–410, October 2010.

[128] George K. Zipf. *The Psychobiology of Language*. Houghton-Mifflin, 1935.