Unreeling Xunlei Kankan: Understanding Hybrid CDN-P2P Video-on-Demand Streaming

Ge Zhang, Wei Liu, *Member, IEEE*, Xiaojun Hei, *Member, IEEE*, and Wenqing Cheng, *Member, IEEE*

***Abstract*—The hybrid architecture of content distribution network (CDN) and peer to peer (P2P) is promising in providing online streaming media services. In this paper, we conducted a comprehensive measurement study on Kankan, one of the leading VoD streaming service providers in China that is based on a hybrid CDN-P2P architecture. Our measurements are multi-fold, as follows. 1) Kankan adopts a loosely-coupled hybrid architecture, in which the user requests are handled by its CDN and P2P network independently. 2) Kankan deploys a small-scale CDN densely in three geographic clusters in China. It adopts specific redirection servers to dispatch the nationwide requests. 3) Kankan adopts a dual-server mechanism to enhance start-up video streaming. It also provides the CDN acceleration in case of inefficient P2P streaming performance. 4) According to our studies on the peer cache lists, the video contents stored in Kankan peers update quite slowly. The average lifetime of cached videos is longer than one week. Our results show that, by utilizing the slow-varying contents cached in peers and deploying various CDN enhancement mechanisms, Kankan provides a large-scale VoD streaming service with a small-scale fixed infrastructure. Insights obtained in this study will be valuable for the development and deployment of future hybrid CDN-P2P VoD streaming systems.**

***Index Terms*—Content distribution network (CDN), hybrid content distribution network peer to peer (CDN-P2P), peer-to-peer (P2P) network, video on demand (VoD).**

I. Introduction

LAUNCHED in 2007, Xunlei Kankan[1] has become one of the leading video-on-demand (VoD) streaming service providers in China with 31.4 million daily unique users in the end of 2012 [1]. In Xunlei Kankan, users are able to enjoy the streaming service for hundreds of thousands of distinct video contents on the Internet including movies, TV episodes and so on. Unlike other existing Internet VoD streaming systems that adopt either content distribution network (CDN) or peer-to-peer (P2P) network for content distribution, Kankan is built upon a hybrid CDN-P2P architecture. A Kankan player requests video segments not only from the CDN servers, but also from the shared caches in peer neighbors. These Kankan clients also form one of the largest P2P networks on the Internet. In a recent measurement study conducted in Jan. 2011, 7.4 million unique peers were observed in one week for just the TV show category [2].

Recently the hybrid CDN-P2P content distribution systems have received increasing attention in both the research community [3]–[7] and in the industry [8], [9]. The P2P approach can provide cost-effective large-scale streaming, while CDN is advantageous to support video streaming with enhanced performance at the cost of the infrastructure. The hybrid CDN-P2P architecture is a promising approach to achieve trade-off between streaming performance and system scalability.

According to their mix of strategies, the hybrid systems can be further classified into two categories [10]: peer-assisted CDN and CDN-assisted P2P. Most of the existing approaches [3]–[9] fall into the first category, in which the P2P network is designed as a complement component for the regular CDN-based content delivery. In these systems, user requests are mainly redirected and served by the CDN, while the P2P network is applied to improve user experience or alleviate the load stress of the CDN. The peer selection process is tightly coupled with the CDN redirection scheme. It is notable that, there are few studies on the performance of real-world hybrid systems, and most of existing studies have been conducted via either trace-driven simulations or pure theoretical analysis.

The measurement study on real-world Internet streaming systems is helpful to understand the architecture design issues and the field lessons may provide useful guidelines to optimize other similar systems. There have been a number of studies on CDN-based VoD systems (e.g. YouTube [11], [12], Netflix [13] and Hulu [14]), P2P-based VoD systems (e.g. PPLive VoD [15], Joost [16]). However, to the best of our knowledge, there have been no previous measurement studies on real-world hybrid CDN-P2P VoD system. This motivates us to conduct a measurement study on the Kankan system.

We are interested in examining the design trade-off in this hybrid CDN-P2P system architecture, especially the cooperation between the CDN and P2P network in Kankan. It is challenging to achieve deeper insights into the Kankan system because the Kankan protocol is proprietary. In order to build the measurement tools that were used to collect much of the data in this paper, we had to analyze a large portion of the Kankan protocol via laborious protocol analysis. In particular, we are able to understand part of the signaling messages and data traffic between the Kankan clients and CDN servers successfully. We conducted a comprehensive measurement study to uncover four specific issues of Kankan system.

Manuscript received June 28, 2014; revised October 07, 2014; accepted December 09, 2014. Date of publication December 18, 2014; date of current version January 15, 2015. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61371080, Grant 61301127, and Grant 61370231. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Pal Halvorsen. *(Corresponding author: Wei Liu.)*

The authors are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: zhangge@mail.hust.edu.cn; liuwei@mail.hust.edu.cn; heixj@mail.hust.edu.cn; chengwq@mail.hust.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2014.2383617

[1][Online] Available: http://www.xunlei.com/

(1) *CDN-P2P architecture*: We carried out a detailed traffic analysis on the encrypted proprietary Kankan protocol, and figured out the basic architecture of Xunlei Kankan. Different from the tightly-coupled CDN architectures proposed in the existing studies [3]–[5], Kankan deploys a loosely-coupled hybrid architecture. There is no unified service entry and user requests are handled by the CDN redirection servers and the P2P trackers separately.

(2) *System deployment*: We deployed a number of distributed crawlers in PlanetLab [2] to measure the Kankan CDN. In the measurement duration of eight months, 293, 023 videos in total were identified and the CDN servers that stored them were located. Only 258 distinct server IP addresses were observed, which are much smaller than the thousands of server addresses observed in YouTube [12] and Hulu [14]. By mapping the IP addresses to geographic locations, the Kankan CDN servers are found to locate densely in the second-tier cities within three geographic clusters, whose centers are in the proximity of three super cities in China. It is known that the Chinese backbone networks are organized following the provincial hierarchy.[3] The deployment of the Kankan CDN does not favor on the major cities and does not spread to every province. It is different from LiveSky, which deploys its CDN servers in almost every province [8].

(3) *Content delivery*: We investigated the Kankan streaming behaviors in various phases and conditions. In the initial streaming phase, two content servers are found in streaming, which is different from the single server streaming in YouTube [12], Netflix [13] and Hulu [14]. When the P2P connections are ready, it works in the mode of hybrid CDN-P2P streaming. If the player finds that the cached video in its local buffer is sufficient for video playback, it will stop streaming from the CDN servers. Once its buffer tends to flush empty, it will stream proactively from the CDN servers again. The Kankan CDN serves as a complement component for the Kankan P2P streaming, and its main role is to enhance the performance at the initial streaming.

(4) *Content distribution*: The cached contents in the Kankan CDN and the P2P network were tracked and compared to understand the video distribution patterns. We collected the lists of the cached videos in 3,085,603 peers during one day, and analyzed the distribution of video segments. The top 20.0% visited movies are cached in more than 5 CDN servers and 844 peers. While the least visited movies are cached in the Kankan CDN only. A novel metric, named *content lifetime*, is introduced to characterize the content dynamics in the Kankan peers. We found that the contents in the peer caches are updated quite slowly. The average lifetime of the cached movies is 209.5 hours, which indicates that the cached videos in the Kankan peers can serve other peers for more than one week.

[2][Online] Available: http://www.planet-lab.org/

[3]The major Chinese ISPs, i.e., China Unicom and China Telecom, organize their backbone networks in a hierarchy as the same as the China's provincial organization [17]. These backbone nodes are usually located in the provincial capitals.

In summary, the CDN and the P2P network in the Kankan system are loosely-coupled. In the content discovery phase, the Kankan client sends requests to the CDN and the P2P network separately. During the content streaming phase, two CDN servers are assigned to enhance the streaming performance in the startup stage. To achieve efficient content distribution and high content availability, the Kankan CDN caches more copies of the most visited contents, and it is also responsible for storing those least visited contents. Benefiting from the slow-varying contents cached in millions of peers and the various CDN enhancement mechanisms, Kankan provides a large-scale smooth VoD streaming service with only a small-scale fixed infrastructure. Insights obtained in our study may be valuable for the development and deployment of future hybrid CDN-P2P VoD streaming systems.

The remainder of this paper is organized as follows. The background and related work are introduced in Section II. In Section III, we present an overview of the Kankan system architecture. The measurement results on system deployment, content delivery and content distribution in Kankan are presented in Section IV, Section V and Section VI. Finally, we conclude this paper in Section VII.

II. Background and Related Work

A. Hybrid CDN-P2P Architecture

CDN has been considered as the main pillar of video distribution over the Internet. CDN servers are geographically distributed in multiple ISPs and push contents close to the end users. With the ever-increasing amount of video traffic, CDN has been facing with several challenges, such as the support for HTTP-based video delivery [18], and the scalability problem [7]. On the other hand, P2P networks provide a cost-effective approach to share multimedia contents among a large number of participating peers. However, due to the peer churn and heterogeneous peer resources, P2P VoD streaming also suffers from various impairments, such as high startup delay [15].

Given the complementary features between CDN and P2P network, some hybrid architectures have been proposed previously. One of the early work was proposed in [3]. The authors analyzed the service handoff issues between CDN and P2P network, and demonstrated the benefits through simulations. Various applications with hybrid architectures were then proposed and studied, including hybrid live streaming [5], [8], [19], hybrid file sharing [4], [9], hybrid video delivery [6] and 3D Tele-immersive contents [20]. A detailed survey can be found in [10].

It is notable that most of the existing studies were evaluated via simulations, whose inputs were the traces either synthesized or collected from other systems. There are only two studies on the real-world commercial hybrid systems. One is LiveSky, a P2P-enhanced live streaming system in China [8]. The other is NetSession, a peer-assisted CDN system deployed by Akamai [9]. Both systems are evaluated based on the system traces. To the best of our knowledge, there have been no previous measurement studies on any large-scale real-world commercial hybrid CDN-P2P VoD systems.[4]

[4]A hybrid VoD system named NextShare was evaluated using the measurement approach in [6]. It was a testbed system supported by the European FP7 project COAST, with only hundreds of peers.

TABLE I
EXISTING MEASUREMENT STUDIES ON CDN/P2P-BASED VOD SYSTEMS

System	Main Tech.	Infrastructure	Content Discovery	Content Delivery
YouTube	CDN	3-tier physical cache hierarchy [12]; server clustered in several U.S. cities [11]; $6,000$ IP addresses observed [12]	DNS-based resolution and HTTP redirection [12]; location-aware server selection [12]	HTTP-based video delivery; single and fixed server assignment [11]
Netflix	CDN	own data centers for video source, Amazon cloud for control, three commercial CDNs for delivery [13], and OpenConnect CDN for traffic reduction [21]	CDN assignment by manifest file in cloud [13]; DNS-based resolution for CDN servers	HTTP-based video delivery; single and fixed server assignment [13]; switch among multiple CDNs [13]
Hulu	CDN	three commercial CDNs [14]; $3,141$ IP addresses and 155 clusters observed [14]	CDN assignment by manifest file [14]; DNS-based resolution for CDN servers	HTTP-based video delivery; single and fixed server assignment [14];
PPLive VoD	P2P	source servers, bootstrap server for tracker assignment, and trackers [15]	three methods (tracker based, DHT, gossiping) [15]	P2P data exchange preferred, source server assistance in insufficient case [15]
Joost	P2P	source servers, channel servers, and trackers [16]	tracker based peers discovery [16]	UDP-based video delivery, P2P exchange with source server stream [16]

B. CDN and P2P-Based Internet VoD Systems

Some measurement studies have been reported on the CDN or P2P-based streaming systems. We summarize the measurements of five existing systems (YouTube, Netflix, Hulu, PPLive VoD[5] and Joost) in the following aspects: 1) infrastructure; 2) content discovery; 3) content distribution. The comparison results are presented in Table I.

The CDN-based VoD systems (e.g., YouTube, Netflix and Hulu) share several common features. They are supported by the large-scale commercial CDNs, which deploy thousands of servers in various ISPs. They rely on the DNS infrastructure to provide request redirection. Videos are delivered based on the HTTP protocol. Once a CDN server is assigned for a video request, it becomes the only source during this video streaming session. Some mechanisms, such as the rate adaptation, are usually adopted to enhance the streaming performance.

Compared with the CDN-based VoD systems, the P2P-based systems (e.g., PPLive VoD and Joost) deploy less infrastructure. Despite the source servers that store the original video files, only dozens of channel servers or P2P trackers are provisioned in the infrastructure. Trackers are responsible for replying the candidate peer swarm to every video request. The video data are usually delivered using the UDP protocol.

These issues are the keys to design Internet VoD system. In Section III, we will compare the Kankan system with the existing CDN or P2P based systems in these issues.

C. User Behavior and Content Distribution in VoD Systems

The characteristics of user behaviors are important in evaluating the performance in traditional VoD systems. There have been some previous measurement studies on the user behaviors in the existing Internet VoD systems, such as PPLive VoD [15], [22] and YouTube [23]. However, in the P2P-based or hybrid VoD systems, another important aspect of user behaviors is the video distribution cached in peers. Since the contents in peer's cache can be shared for the future requests, the content availability in peers plays an important role in a large-scale P2P VoD streaming system. A detailed characterization of the content distribution in peer's caches is helpful to quantify the benefits of the P2P network in the hybrid VoD systems.

[5]PPLive was renamed as PPTV in Dec. 2009. In order to be compatible with the previous literature, we use PPLive in this paper.

To date, there have been few measurement studies on the content distribution in peer caches except [15]. With the traces from PPLive VoD, Huang *et al.* focused on three-level movie objects, and investigated the number of users owning different segments of these movie objects. By comparing the trace-driven simulation results against the real request records, they evaluated the replication policy in PPLive VoD. In Section VI, we will investigate the cached contents in the Kankan P2P network, and present our analysis the content dynamics in peer caches.

III. KANKAN VOD STREAMING SYSTEM

A. Brief Introduction on Xunlei Kankan

Xunlei Limited is one of the top 10 largest Chinese Internet companies. One Xunlei's major services is to provide P2P-based file acceleration for the end-users. The P2P-based Xunlei accelerator was reported to have close to 300 million monthly users for the three-month period that ended on Mar. 2014 [24]. Another major service provided by Xunlei is the Kankan VoD service. The Kankan player was initially produced as a value-added service for Xunlei accelerator in 2007. Afterwards, Kankan became a stand-alone VoD system, which had its own media player software and web site (www.kankan.com). According to the iResearch, Kankan was reported to have about 31.4 million daily unique users in 2012 [1], which was close to the 34 million daily visits to PPLive in 2012 [25].

Despite the popularity of Xunlei and Kankan in China, it has received relatively little attention in the research community to date. Unlike PPLive which was open in providing system traces [15] or sharing system experience [22], not much about Kankan has been disclosed by Xunlei company itself. On the other hand, the encrypted proprietary protocols adopted in the Kankan system make it difficult to reveal the system designs without reverse engineering.

In our previous work on Xunlei measurement [2], we conducted laborious protocol analysis and understood a large portion of the Xunlei protocol. We also obtained some preliminary results on the Kankan P2P network network: 1) The Kankan client adopts the same Xunlei P2P protocol with Xunlei accelerator; 2) 7.4 million unique Kankan peers were measured in a week for just the TV show category.

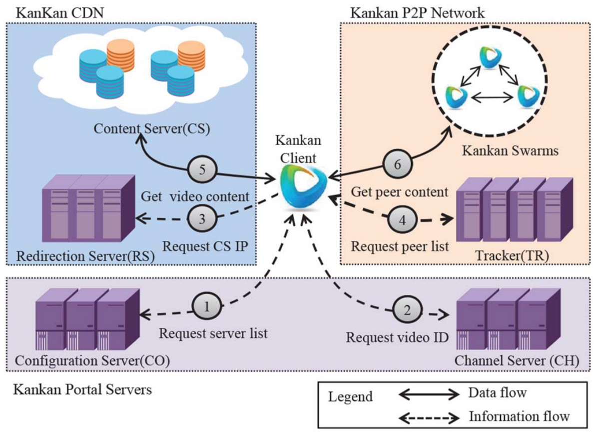


Fig. 1. Hybrid CDN-P2P VoD streaming architecture of Kankan.

B. System Architecture of the Kankan VoD System

We performed an in-depth protocol analysis on the Kankan system. Our analysis consisted of two stages. First, we utilized the packet capture tool Wireshark to capture all the downlink and uplink traffic of the local Kankan client, and then performed the protocol analysis on those captured packets. The clues that could be utilized include the IP addresses, the repeated ID, the domain names, and so on. Then, we can identify the signaling messages, major components and their interactions in the Kankan system. In the second stage, we discovered the major signaing procedures of the Kankan content discovery. The signaling interaction between the Kankan client and other Kankan servers or peers were tracked and analyzed.

Through our measurement study and protocol analysis on the Xunlei Kankan streaming system, we have gained significant insights into the protocols and streaming mechanisms of its hybrid streaming architecture as shown in Fig. 1. The system consists of three major components including the Portal Servers, the Kankan CDN and the Kankan P2P Network. The Kankan client, which is located in the center, queries various servers following specific procedures. We will discuss these components in details as follows.

1) Kankan Client: The Kankan client is not only a VoD streaming player but also a peer in the Xunlei P2P network.

- *Media player*: As a streaming player, the client downloads the video segments to the local disk buffer during the streaming process. The buffered contents will be updated automatically by the cache replacement algorithms of the client, or deleted manually by the user.
- *Kankan Peer*: The streaming engine serves as a peer in the Kankan P2P network. Each peer has a unique ID. This ID consists of two parts: the first 12 bytes are the hexadecimal equivalent to the MAC address; the remaining 4 byte are generated during the software installation.[6] With this ID, every client can be uniquely identified in the Kankan P2P network.

[6]We observed that this client ID was changed when the Kankan client software was reinstalled on the same computer.

TABLE II
SERVERS IN KANKAN VOD SYSTEM

Subsys.	Servers	Abbr.	Main Function
Portal	Configuration Serv.	CO	Reply server addresses to clients
	Channel Server	CH	Map video titles to video IDs
CDN	Content Server	CS	Store video and stream to clients
	Primary CS	PCS	Store the complete video copy
	Secondary CS	SCS	Store the partial video copy
	Redirection Serv.	RS	Assign CS for requested videos
	Primary PS	PRS	Reply PCS for specific videos
	Secondary RS	SRS	Reply SCS for specific videos
P2P	Tracker Server	TR	Assign peers for requested videos

2) Kankan Servers: The Kankan architecture deploys various servers, whose functions are listed in Table II based on our measurements.

- *Configuration servers (COs)*: They have the domain name "conf.xmp.kankan.com". A CO stores the other servers' domain name or IP address.
- *Channel servers (CHs)*: They have the domain name "pianku.xmp.kankan.com". CHs store the video information, including the title and the ID.
- *Content servers (CSs)*: They store the video copies and serve as the streaming servers. According to their policies in video caching, we classify them into two subtypes, Primary CS (PCS) and Secondary CS (SCS). As we will report in Section V, a PCS stores the complete video copy, while a SCS only stores the partial copy.
- *Redirection servers (RSs)*: They are responsible for redirecting video requests to candidate CSs. After initialization, a Kankan client knows two types of RS addresses, "cl.kankan.xunlei.com" and "web.cl.kankan.xunlei.com", which will redirect video requests to PCS and SCS, respectively. As a result, we name these two types of RSs as Primary RS (PRS) and Secondary RS (SRS).
- *Trackers (TRs)*: They store the list of the cached video reported by the peers. They have three domain names, "mvhub5pr.sandai.net", "hub5pr.sandai.net" and "hub5p.sandai.net".

3) VoD Streaming Process in Kankan: The signaling procedures between the Kankan client and various servers are illustrated by different arrows in Fig. 1, upon which the sequence numbers are marked. Once being installed, the Kankan client will actively request the list of the nearest Kankan servers from the pre-configured CO (step 1). When the user selects one video to play, the client will request the video ID from the known CH (step 2). Knowing the video ID, the client will query both PRS and SRS for the candidate CSs (step 3). It will also query the TR for the list of candidate peers (step 4). The replied CDN PCS and SCS will stream to the client upon the requests (step 5). After joining the P2P swarm of the requested video, the client can also exchange video data with other peers in the same swarm (step 6). Finally, the client can download video contents from both the Kankan CDN and the Kankan P2P network simultaneously.

We are interested in understanding the cooperations between the Kankan CDN and the P2P network, especially the joint work of the two subsystems in content discovery and content delivery. In the following sections, we investigate these issues in detail.

TABLE III
TYPICAL KANKAN MESSAGES IN CONTENT DISCOVERY

No.	Time(s)	Src.	Dest.	Protocol	Encrypt.	Part of the message
1	38.9514	Client	SRS	HTTP	No	... GET/getCdnresourceflv&?gcid= 580137262C9B1308CBEE9FE83F7A62EEAE835607 HTTP /1.1 ...
2	38.9516	Client	PRS	HTTP	No	... GET /getCdnresourcene/580137262C9B1308CBEE9FE83F7A62EEAE835607 HTTP /1.1 ...
3	38.9525	Client	TR	TCP	Yes	... 1A5D390B0913BA7CDA4D1738A00634DBAC7C8CFB ...(12Bytes)... 580137262C9B 1308CBEE9FE83F7A62EE AE835607 ...
4	38.9726	SRS	Client	HTTP	No	... var jsonObj = {cdnlist1:[{ip:180.153.91.31, port:8080...}, {ip:125.78.247.142, port: 8080... } ...
5	38.9797	TR	Client	TCP	Yes	... 14DAE974D36DTL6Q|111.123.195.216|12766 ...
6	38.9879	PRS	Client	HTTP	No	... var jsonObj = {cdnlist1:[ip:125.78.142.164,port:8080 ...]} ...

* The video titled "Make Progress Every Day" was requested from our lab in Wuhan in 01:02:38 AM, Mar. 2013.

C. *Content Discovery in Kankan VoD*

After obtaining the video ID, the Kankan client is ready to search the video content from both the Kankan CDN and the P2P network. One of our questions is whether there exists some coordination mechanism to schedule the resources in CDN and P2P network. We conducted several experiments to investigate it. One typical set of results are listed in Table III.

This experiment was conducted from our lab in Mar. 2013. By conducting the protocol analysis, we decoded the signaling messages between the Kankan client and other servers. The content discovery related messages are provided in Table III. There are three connections in total from the client to PRS, SRS and TR respectively. The communications with PRS and SRS utilize HTTP; while those with P2P TR are based on the encrypted TCP messages. We analyze these messages in the following issues.

1) Content Discovery Procedure: Focusing on the message time, we observe that the three requests were sent out almost at the same time. The tiny time difference may be due to the DNS resolution. We observed similar results in other experiments. Thus, we conclude that the Kankan client performs content discovery in the CDN and P2P network simultaneously, or, there is no fixed sequence for step 2 and step 3 in Fig. 1.

2) Video ID in Content Discovery: The video ID appears repeatedly in the requests to PRS, SRS and TR. In the No. 1 and No. 2 messages, a 20-byte repeated hex value (marked by wavy underline) is found. It turns out to be the file ID used in the Kankan CDN, and is denoted as f_{id}^{cdn}. In the No. 3 message, f_{id}^{cdn} also appears. As reported in the previous work on Xunlei [2], a 20-byte hash value is adopted to identify the file in Xunlei P2P network. This design is also confirmed here. The hash value is marked by underline, and is denoted as f_{id}^{p2p}. We also find a 12-byte file size exchanged during content discovery. Thus, the complete video ID in the Kankan P2P network can be denoted as $[f_{id}^{p2p}, f_{size}, f_{id}^{cdn}]$. With this video ID, a Kankan client is able to request the same video content from both the CDN and the P2P network separately.

3) Content Servers in Content Discovery: In the No. 4 and No. 6 response messages, the Kankan client received the responses from SRS and PRS with two and one IP addresses respectively. When the client receives multiple addresses from PRS or SRS, it just selects one as the PCS or SCS for the requested video. In this experiment, the CS with address "125.78.142.164" was taken as PCS, and the CS with "180.153.91.31" was taken as SCS.[7]

4) Peers in Content Discovery: In the No. 5 response message, the P2P TR replies with a number of candidate peers. The number of peers may vary from zero to several hundreds. We demonstrate one entry of the candidate peer in the message part in Table III. It consists of three parts: a 16-byte peer ID, the peer's IP address and the selected port.

D. *Content Delivery in Kankan VoD*

Once the Kankan client establishes connections to the Kankan CSs and peers, it initiates the video streaming process as follows.

1) Streaming Protocol in Kankan: The Kankan CDN adopts the unencrypted HTTP protocol in both the signaling exchange and the video streaming. The Kankan P2P network adopts the encrypted TCP-based protocol for signaling between TRs and the clients. It adopts its proprietary UDP-based protocol for the peers to exchange their data.

2) Video Caching in Kankan Clients: After the Kankan client is installed at the local host, a video cache with the default size of $2GB$[8] will be allocated in the local hard disk. If the user selects one video title for playback, the client will first play a fixed-duration advertisement video (about 45 seconds). During this period, the client will actively request the targeted video data from both CDN and P2P network, and store the downloaded data into the local cache. Once the video starts playback, the client will prefetch more video data before the current playback point.

3) P2P Data Exchange Between Kankan Clients: The Kankan client periodically contacts with the closest TR, reporting its online status (about every 5 seconds) and the list of the cached videos (about every 2 minutes). Thus, the Kankan TRs are able to track the peer swarms for various videos. Once receiving a video request, a TR will reply with a candidate peer list, which is a random subset of that video swarm. The client will set the candidate peers as its neighbors. It exchanges the video bitmap with its neighbors, and requests the needed data from those neighbors who already have them. Meanwhile, the client also upload video data to other peers upon their requests.

[7]The PCS and SCS in this experiment are located in two cities, Putian and Shanghai, which are far away from the city of request origin, Wuhan.

[8]This cache size is configured by the Kankan client during the software installation. This value can be found in the software configuration, for example, in the system registry of the Windows OS.

4) Hybrid CDN-P2P Streaming: The joint work of CDN and P2P can be described by three streaming modes: pure CDN streaming, hybrid CDN-P2P streaming and pure P2P streaming. Since it requires time for the client to setup connections with other peers, the Kankan CDN plays a major role in the start-up stage of video streaming. In most of the cases, the client starts first in the pure CDN streaming. Once the client receives video data from peers, it transits into the hybrid CDN-P2P streaming. When the total P2P traffic is sufficient for video playback, the client will cease CDN streams and then it jumps into the pure P2P streaming. If the P2P streaming traffic becomes insufficient, the client will restart CDN connections and return to the mode of the hybrid CDN-P2P streaming.

E. Summary

Our main observations on the Kankan infrastructure, content discovery and content distribution can be concluded as follows.

- *Kankan portal*: The functions of Kankan portal servers are similar to those in Netflix [13] and PPLive VoD [15], which also have the configuration and channel servers.
- *Kankan CDN*: The redirection mechanism of Kankan CDN is different from that in the CDN-based VoD systems, which rely on the DNS infrastructure to redirect video requests [12]–[14]. Compared with them, the design of the stand-alone redirection servers in Kankan is a light-weight implementation.
- *Kankan P2P network*: The peer discovery process in Kankan is also different from some P2P-based VoD systems. For example, PPLive VoD adopts bootstrap to assign trackers to the clients [15], while the Kankan client can directly request the known trackers for the candidate peer list, which is simpler in the peer process.
- *Loosely-coupled architecture*: There is no unified service entry for discovering content in Kankan. User requests are handled by the redirection servers in the CDN and the trackers in the P2P network separately. We conclude that the Kankan CDN and the P2P network are loosely-coupled. This is different from the other tightly-coupled hybrid CDN-P2P schemes proposed in [3]–[6].
- *Dual-server streaming*: For every user request, two CDN content servers are assigned to the Kankan player. We call this as a dual-server streaming mechanism. This is also different from the mechanisms in YouTube [12], Netflix [13] and Hulu [14], in which only one server is assigned for each video streaming session.
- *Client-driven hybrid streaming*: There is no back-end scheduler observed for the hybrid CDN-P2P streaming in Kankan. The switch of CDN streaming is initiated by the client based on the local streaming status.

IV. Deployment of the Kankan Infrastructure

In this section, we investigate where and how Kankan deploys its content servers, and its server redirection mechanism.

A. Measurement Methodology

As shown in Fig. 1, the Kankan client can request the addresses of RSs during initialization. By requesting the RSs, the client can obtain the addresses of CSs. To harvest all the CSs, two key issues have to be addressed.

The first issue is how to collect all the video IDs. The video meta data are stored in the CHs. However, due to the protection mechanism against Denial-of-Service (DoS) deployed in Kankan, a crawler, which sends a large number of requests to the CHs, will be easily identified as a spammer and be blocked. Fortunately, the video IDs can also be found in the Kankan web site. Thus, we developed a web crawler to collect them.

The second issue is how to collect all the IP addresses of the CSs. We implemented a CS crawler, which can send video requests to the RSs and collects the distinct content servers from the responses. The commercial CDNs are usually deployed with locality-aware mechanisms [12], [26]. In order to harvest the complete set of Kankan CSs, we utilized the distributed PlanetLab platform, and deployed this CS crawler on 413 PlanetLab nodes. We also developed a crawling task controller to dispatch the tasks and collect results from these distributed crawlers.

TABLE IV
Servers in the Kankan Infrastructure

Subsystem	Servers	Domain Name	IPs	Sum
Portal	CO	conf.xmp.kankan.com	8	16
	CH	pianku.xmp.kankan.com	8	
CDN	PRS	cl.xunlei.kankan.com	6	266
	SRS	web.cl.xunlei.kankan.com	2	
	CS	N/A	258	
P2P	TR	mvhub5pr.sandai.net	6	18
		hub5pr.sandai.net	8	
		hub5p.sandai.net	4	
Total			300	

B. Servers in the Kankan Infrastructure

The measurement experiments on the Kankan infrastructure were conducted from 1 September 2011 to 30 April 2012. For the Kankan servers with known domain names, we recorded the appeared IP addresses after DNS resolution. For the CDN CSs, we firstly utilized the web crawlers to collect video IDs. Total 293,023 distinct video IDs were collected. These contents cover all kinds of videos in Kankan, such as movies, TV episodes, cartoons and entertainment shows. Then, we utilized the distributed CDN crawlers in PlanetLab nodes to request all these videos. The Kankan servers found in our measurement are listed in Table IV.

In Table IV, for each type of the Kankan servers, the measured number of IP addresses are tabulated. Total 300 IP addresses are found for the servers in the Kankan infrastructure, among which 258 addresses are identified as CDN content servers. Note that it is possible that multiple physical servers are behind each IP address. Hence, the number of IP addresses here can only be used as an approximate number of the actually deployed servers. Compared with the CDNs of YouTube ($6,000+$ IP addresses [12]) and Hulu (3,141 IP addresses [14]), the scale of the Kankan CDN is much smaller. This result motivates us to investigate the geographic deployment of CDN content servers.

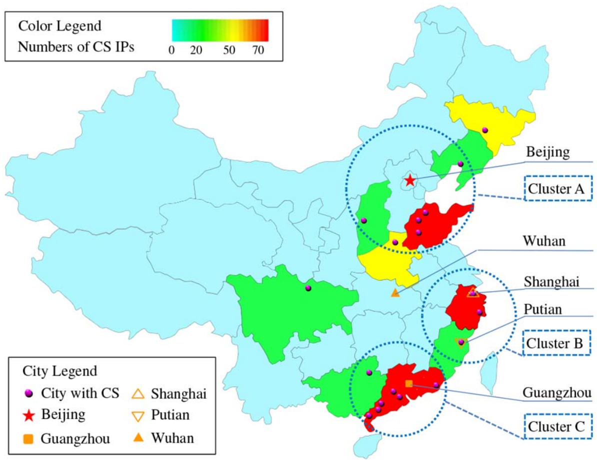


Fig. 2. Geographic deployment of the Kankan CDN servers.

C. Geographic Deployment of the Kankan CDN

We mapped the IP addresses of content servers to a region by querying the free MaxMind GeoIP database. All the CSs were found to be located within mainland China. We further mapped the IP addresses by querying the Chunzhen database (QQWry)[9] for the city-level location. The results are quite interesting. The 258 addresses are densely located within 18 cities in 10 provinces. It is known that province and city (or county) are the two levels in the Chinese administrative hierarchy. There are total 34 provinces and 333 second-tier cities in China.[10] As a VoD service provider serving nationwide users, Kankan only deploys its CSs in 5.4% of the total Chinese cities.

We further investigate this geographic deployment by counting the CSs into the province level. The resulted geographic distribution of Kankan CSs is presented in the map of mainland China in Fig. 2. The sum of CSs in each province is indicated by different background colors, ranging from green to red. The province that has less than 20 CSs IP addresses is filled with green, while that with more than 50 is filled with red. The major cities deployed with content servers are also marked with the purple points.

As shown in Fig. 2, the Kankan CDN is deployed in geographic clusters. Three clusters are found in the centers of Beijing, Shanghai and Guangzhou, which are the three most developed super cities in China. We mark these clusters by dashed circles and name them cluster A, B and C, respectively. As reported in [17], the major Chinese backbone networks are organized following the provincial hierarchy. Kankan does not prefer server deployment in the provincial capitals. There is no CSs deployed in some provincial capitals or even super cities, such as Beijing. This is different from other streaming systems that serve nation-wide users, such as LiveSky, which deploys the CDN servers in every province in China [8].

We wonder whether such a selective cluster-like deployment can meet the users' demands. We classify the cities in clusters into three levels: the centric super city in that cluster, the provincial capitals and the rest second-tier cities. We count the content servers at different city levels. On the other hand, we collect the data of the netizen population published by CNNIC [27] in 2012, and use the data as the reference for the potential Kankan users in different clusters. The results are present with the format of "x/y" in Table V, in which x stands for the number of cities deployed with content servers, and y stands for the total cities in that cluster.

TABLE V
DEPLOYMENT OF KANKAN CSS IN DIFFERENT CITY LEVELS

Cluster	Center	Super	Capital	Second	Netizens
A	Beijing	0/1	2/6	4/65	24.2%
B	Shanghai	1/1	0/2	2/28	18.5%
C	Guangzhou	0/1	0/1	7/33	19.1%
Sum		1/3	2/9	13/126	61.8%

As shown in Table V, the CSs are only deployed in 1 super city, 2 provincial capitals and 13 second-tier cities. Most of them are not deployed in the first-tier (super or capital) cities. Every cluster serve around 20% of the total netizens. Thus, although the servers are not located in first-tier cities, they are still deployed close to 61.8% of the potential VoD subscribers.

D. Server Redirection Mechanism

If there are some Kankan CSs deployed in one province, the users in that province will be possibly served by this local CS. For the users outside any cluster, their experience will be affected by the request redirection decisions by Kankan. We are interested in quantifying the redirection for the users from different locations. Several experiments were conducted for this concern.

We selected two vantage nodes of PlanetLab to test the Kankan RSs: one is located in Guangzhou, the super city in cluster C (marked with ■ in Fig. 2); the other is in Wuhan, which locates in the center of China and outside of any cluster (marked with ▲ in Fig. 2). We configured the CDN crawler in the two nodes to query the RSs for the CSs. We also developed a tool to measure the latencies from the vantage nodes to the replied CSs. The latency was measured using ICMP Ping. Considering the possible network dynamics, we conducted the measurements every hour from 9:00AM to 5:00PM (usually the less congested period), and took the average of the original latency measurements as the final result.

Redirection for Nodes Inside and Outside Clusters: In the first set of redirection experiments, we compared the redirection results when the crawler requests from Guangzhou and Wuhan. We selected 1,000 videos randomly from the collected video set, and scheduled the crawlers to request these videos. The results of the redirected CSs' locations are tabulated in Table VI.

TABLE VI
REDIRECTION RESULTS FOR DIFFERENT ORIGINS

Request Origin		Redirection Results		
City	Cluster	Cluster A	Cluster B	Cluster C
Guangzhou	Cluster C	0%	17.9%	82.1%
Wuhan	none	34.6%	39.7%	25.7%

[9][Online] Available: http://www.cz88.net/

[10]Constitution of the People's Republic of China, http://www.hrol.org/Documents/ChinaDocs/Obligations/2012-11/255.html

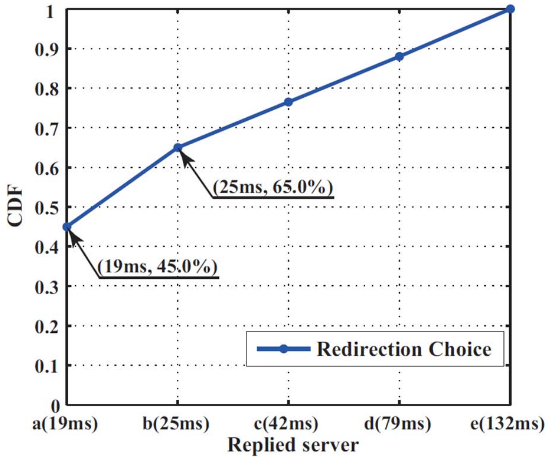


Fig. 3. CDF of the replied CSs when requesting from Wuhan.

As shown in Table VI, when requesting from Guangzhou, 82.1% of the requests are handled by the local cluster C, and 17.9% are forwarded to the nearby cluster B. When requesting from Wuhan, the requests are almost evenly dispatched to the three clusters. We conclude that the Kankan redirection mechanism deploys load balancing for the requests.

We further investigated the difference of the redirection results when requesting from Guangzhou and Wuhan. Studying the measured latencies from the vantage nodes to the replied CSs, we found that the Kankan tried to redirect requests to the "nearest" CS with the smallest latency. When requesting from Guangzhou, 73.9% of the requests were forwarded to the nearest CS in cluster C. When requesting from Wuhan, 64.6% of the replied CSs were the closest to Wuhan in their cluster. Compared with the users outside clusters, the users inside exhibit 9.3% more bias to be served by the nearest CS.

Redirection for Nodes Outside Any Clusters: In the second set of redirection experiments, we investigated the worst case of the redirection results for the users outside of any cluster areas. We generated a heavy-load request trace, in which the same video was requested for 200 times continually in every 5 seconds. The crawler in Wuhan was configured to request the RSs with this trace. We recorded the replied CSs as well as the latencies to them. The CDF of the replied CSs in the rank of latencies are plotted in Fig. 3.

Totally 5 CSs were found in this experiment, and we mark them with a to e according to their latencies to Wuhan. The nearest CS a is with the latency of 19 ms, while the second nearest b is with 25 ms. 45.0% of the total requests is redirected to a, and 20.0% to b. The rest of the requests are almost evenly distributed among the server c, d and e. We infer that there are at least two CS levels in the Kankan redirection mechanism. The nearest CS is in the level with the first priority, which handles nearly half of the total requests. In case of heavy request loads, the requests will be evenly dispatched among the CSs at the least priority level, even if the latency is as high as 132 ms.

E. Summary

Our main measurement results on the Kankan CDN are summarized as follows.

- *Scale of the Kankan CDN*: Kankan deploys a relatively small scale infrastructure. Only 258 distinct IP addresses are observed for the content servers in the Kankan CDN, which is much smaller than the thousands of IP addresses reported in the CDN-based VoD system, such as YouTube [12] and Hulu [14].
- *Geographic deployment of the Kankan CDN*: The Kankan content servers are deployed in geographic clusters, which are around the three super cities in China, i.e., Beijing, Shanghai and Guangzhou. This is quite different from LiveSky, which deploys CDN servers in every province in China [8].
- *Performance of the Kankan CDN redirection*: Kankan provides a server redirection mechanism to dispatch the user requests to the content servers. When handling the requests from the users outside any cluster, such mechanism can dispatch the requests evenly to the three clusters of servers. Nearly half of the requests will be redirected to the nearest content server, even in case of heavy request loads.

V. Hybrid CDN-P2P On-Demand Streaming

In this section, we investigate the hybrid streaming mechanisms in Kankan. We are interested in understanding the roles of the Kankan CDN, thus we select two cases when the P2P streaming capacity may be insufficient. First, during the startup phase, it takes some time for the Kankan client to set up initial connections with other peers. Second, under the adverse or dynamic network conditions, the P2P streaming may become insufficient to support smooth playback.

A. Measurement Methodology

In order to identify various Kankan streaming flows, we captured the Kankan traffic and computed the average streaming rate. To investigate the Kankan performance under dynamic network conditions, we implemented a traffic shaper using the Linux Network Emulation functionality (netem) with the Token Bucket Filter (TBF). It was installed on the network gateway in our lab to control the bandwidth of the specific CDN or P2P streams.

B. Hybrid CDN-P2P Streaming in the Startup Phase

As described in Step 4 to Step 6 in Fig. 1, in the startup streaming phase, the Kankan client requests video contents from PCS and SCS. Meanwhile, it also requests the video from the P2P network. We report one set of measurement results in the following: We selected one movie with the 480P quality as the streaming objective and captured the streaming traffic for 600 seconds. To study the CDN assistance, we actively changed the playback point by jumping to the halfway point of this movie at 500 second. We identified the flows from the PCS, SCS and the P2P network, and obtained the throughput results shown in Fig. 4.

As shown in Fig. 4, the flows from the PCS and SCS started from the beginning and stop at around 155.2 second with the average throughput of 198.2 kbps and 88.8 kbps, respectively. The P2P flows started from 6.2 second and streamed with 527.3 kbps. In this experiment, the Kankan CDN only served for a short duration with the total average throughput of 287.0 kbps.

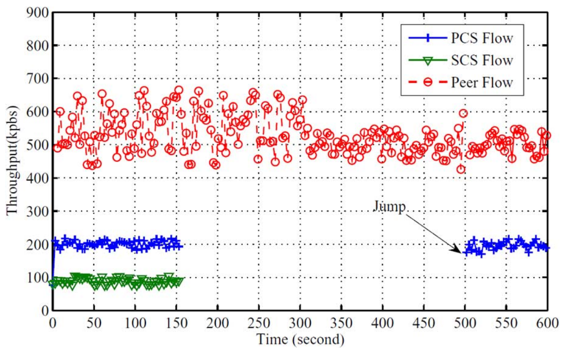


Fig. 4. Hybrid CDN-P2P streaming in startup phase.

After the peers were connected, the P2P flows contributed most of the streaming traffic. At time 500 second we jumped the playback to the halfway point of the movie; then, the P2P streaming became insufficient. The client restarted to stream from the Kankan CDN; however, only the PCS was observed in this streaming session. In the experiments on streaming other movies, we observed similar behaviors of PCSs and SCSs. Thus, we conclude that the SCS only stores the initial part of video contents, while the PCS stores the complete copy.

We also quantify the streaming assistance of the Kankan CDN. As mentioned in Section III, every common user is required to watch a fixed duration of the advertisement video. We denote this advertisement duration as T_{AD}, which is 45 seconds in the current setting of the Kankan client. We denote the average rate of PCS and SCS flows as R_{PCS} and R_{SCS}, and the normal video playback rate as R_{Play}. For the video in this experiment, R_{Play} is about 578 kbps. If P2P streaming is not available, the video contents supplied by the CDN can be as long as $(R_{PCS} + R_{SCS}) \cdot T_{AD}/R_{Play} \approx 22.4$ seconds. In most cases, this duration is long enough for the clients to setup sufficient number of connections with other Kankan peers.

C. *Roles of PCS and SCS in Kankan Streaming*

We have observed the difference in content storage between PCS and SCS. We are interested in understanding their roles in content distribution. As discussed in Section III, the IP addresses of PCS and SCS can be harvested by the crawler to query PRS and SRS. We used the same method and investigated the deployment of PCSs and SCSs in the Kankan CDN. The measurement results are reported in Table VII.

As shown in Table VII, the Kankan content servers have a dual role for different videos. We find that 100% of the CSs serve as PCS for specific videos, while only a proportion (12% $\sim$ 22%) of them are assigned to serve as SCS. For example, among the total 98 CSs in cluster A, only 12 of them serve as SCS and store the initial part for some videos. 15.9% of the total CSs serve as SCS, but none of them are located outside the three clusters.

It should be pointed out that, the term of "cluster" used in this paper only describes the servers' relationship in terms of geographic distance, and is not related to the serving area. In one experiment, we observed that a SCS in cluster C streamed to the vantage nodes in cluster A and B.

TABLE VII
ROLES OF PCS AND SCS IN THE KANKAN CDN

Geographic Area	Total CS	Roles of CS			
		PCS		SCS	
Cluster A	98	98	100%	12	12.2%
Cluster B	66	66	100%	13	16.7%
Cluster C	73	73	100%	16	22.0%
Outside clusters	21	21	100%	0	0%
Sum	258	258	100%	41	15.9%

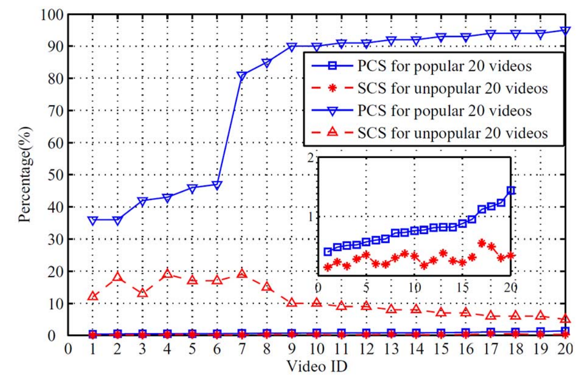


Fig. 5. Streaming traffic proportion for popular and unpopular videos.

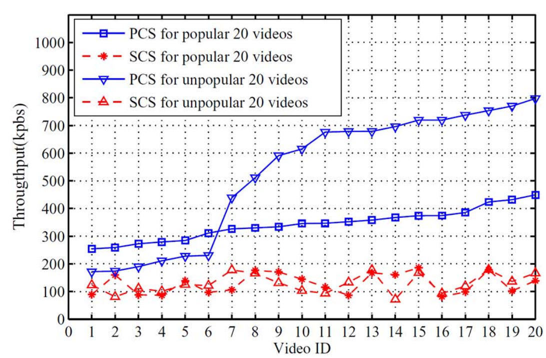


Fig. 6. Streaming throughput for popular and unpopular videos.

D. *Streaming Performance From PCS and SCS*

In order to investigate the performance difference between the PCS and SCS flows, we selected the same set of videos, and measured the streaming throughputs from the PCSs and SCSs to the Kankan clients in Wuhan. In total, we selected 20 popular and 20 unpopular videos in the 480P quality based on the visit records shown in the Kankan web site. We captured the streaming traffic for the first 1000 seconds of each video. The traffic proportions contributed by the PCS and SCS flows are plotted in Fig. 5.

For the popular videos, the highest contribution provided by PCS is 1.5%, and that by SCS is 0.5%. In other words, the P2P traffic contributes at least 98.0% of the video traffic to clients. But for the unpopular videos, CDN traffic contributes the majority. Average 76.3% and 11.1% of the traffic are contributed by the PCSs and SCSs respectively. For the most unpopular 14 videos, there is even no P2P traffic observed at all.

We further investigated the streaming throughputs from these CDN content servers. The measurement results are plotted in Fig. 6.

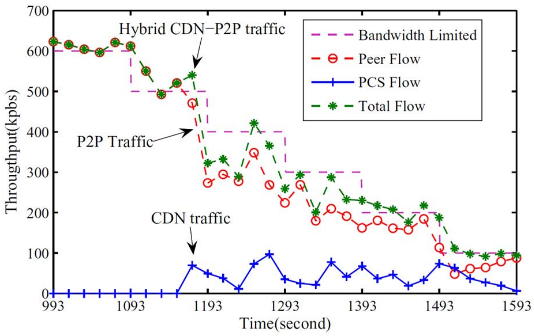


Fig. 7. CDN assistance in case of insufficient P2P streaming capacity.

The PCSs usually stream at higher rates than the SCSs. For the popular videos, the average streaming throughput from PCSs vary from 254.1 kbps to 449.4 kbps. For the unpopular videos, the throughput vary from 171.4 kbps to 797.4 kbps. The PCS streaming throughput of the unpopular videos (e.g., from No. 11 to No. 20) are almost doubled compared with those of popular videos.

On the other hand, the streaming throughput from the SCSs maintain almost the same with the average of 128.8 kbps, regardless of the video popularity. We wonder whether all the SCSs have the smaller connection bandwidth to the clients located in our lab. After checking the IP addresses, we find that some of the SCSs share the same IP addresses of some PCSs. These SCSs are equipped with good enough network bandwidth and similar server capacity with the PCSs in the same addresses. We conjecture that the SCSs are configured with some streaming rate threshold, which are designed to protect the service quality provided by the PCSs.

E. CDN Switching in Case of Insufficient P2P Traffic

In the ideal cases the Kankan client streams video in the pure P2P streaming mode, and the CDN streaming is not required. If the client detects that the video contents cached locally are not sufficient for smooth video playback, it proactively requests the CDN CSs to restart streaming. There are many possible reasons causing the shortage of P2P traffic, such as the peer churn and the dynamic changes of network bandwidth. We are interested in the behaviors of the Kankan CDN assistance. Thus, we emulate continuous adverse conditions of P2P streaming, and investigate the reactions of the Kankan CDN in several experiments.

We report one typical set of results of our experiments. One high-quality video was selected as the examination object, whose average playback rate was around 1008 kbps. We configured one Kankan client to playback this video. When the video streamed to its halfway point, we began to reduce the network bandwidth via the traffic shaper. The bandwidth was reduced by 100 kbps in every 100 seconds. We identified and measured the flows from the CDN servers and the P2P peers, respectively. The experiment results after 993 second are plotted in Fig. 7.

When the total bandwidth was reduced, the received P2P traffic also decreased. As shown in Fig. 7, the client did not request CDN servers immediately when the P2P streaming throughput became less than the playback rate. We conjecture that it first consumed the downloaded contents stored in its local cache. When the cached content became less than a certain threshold, the client restarted the CDN streaming. This switching event occurred at the time 1169.0 second. With the assistance of the Kankan CDN, the total received streaming throughput reached almost the bandwidth limit in the remainder experiment.

We also estimated the cache threshold of the Kankan client to restart CDN connection. We denote the content transmitted from the PCS, SCS and the P2P network as C_{PCS}, C_{SCS} and C_{P2P}, respectively. Then, the local cache size could be estimated as $(C_{PCS} + C_{SCS} + C_{P2P})/R_{Play} = 1179.0$ seconds, which is about 10 seconds longer than the actual playback point of 1169.0 second. We performed similar experiments for other 10 videos, and obtained similar results. Thus, we conclude that there is a threshold for the Kankan client to restart the CDN streaming, which is about 10 seconds ahead of the current playback point.

We wondered whether Kankan enables multiple CDN flows at adverse network conditions. We conducted several similar experiments to examine the PCS flows. We configured the firewall software on the LAN gateway to block all the P2P traffic, and reduced the total network bandwidth by 100 kbps every 100 seconds, starting from an initial value of 500 kbps. We found that there was no newly established CDN connections. The client kept requesting to the PCS which was assigned initially by the RS. We conducted another 10 experiments for different videos and obtained the same results. The Kankan client appears to connect to the same server which is dispatched at the beginning of this video session. We conclude that the Kankan server redirection mechanism only works in the startup phase of video streaming.

F. Summary

We summarize the main measurement results on the hybrid streaming in Kankan as follows.

- *Contributions of CDN and P2P network in streaming*: In the initial streaming phase, two CSs are assigned to a Kankan client. The CDN streaming ceases after the P2P streaming provides a sufficient download rate. For the popular videos, the P2P streaming contributes up to 98.0% of the video content. For the unpopular videos, only the CDN servers are responsible for streaming the video contents.
- *Different server roles in the dual-server streaming*: The dual-server mechanism is implemented by assigning CSs with different roles for various videos. The PCS stores the full version and plays the major role in streaming. Nevertheless, the SCS only stores the initial part of the video and will not join the later hybrid streaming.
- *CDN-assisted hybrid streaming*: The Kankan CDN provides assistance in case of insufficient P2P streaming. If the video content cached locally is less than 10 seconds ahead of the current playback point, the client will restart the CDN streaming. The client only requests to the assigned PCS or SCS, and no additional CDN servers are provided by Kankan.

VI. Content Distribution in CDN-P2P Network

We are interested in quantifying the "storage capacity" of the Kankan P2P network compared with the CDN. First, we report the overall measurement on videos in Kankan. Then, we investigate the following two issues: how many CDN servers and peers are involved in storing one video content, and how long can the peers store it.

A. Measurement Methodology

It is known that the caching policies in CDN and P2P network are quite different. The CDN servers may proactively replicate the videos based on the popularity or other metrics, while the peers will reactively cache and replace the downloaded videos. The content availability largely determines the utilization of these two Kankan subsystems. We are then motivated to investigate the content distributions of the same video within the two Kankan subsystems.

As to the Kankan CDN, we applied the same crawling method in Section IV-A to track the content distribution in the CSs. Our crawler collects a candidate set of video IDs, requests to Kankan RSs, and records the replied CSs. In the Kankan P2P network, we applied a similar crawling method. Our crawler sends video requests to the Kankan trackers, and records the replied peer lists. Note that peers may join and leave the video swarm dynamically. In order to obtain the "snapshot" of the Kankan P2P network, we develop a P2P crawler to support continuous crawling in a short time. In our measurements, the process of crawling the candidate peer lists of all the movies takes about 20 minutes, which does not trigger the DoS attack alert by the trackers. The harvested peer lists are stored in our database for further analysis.

B. Videos in Kankan

We selected the categories of movie and TV show in our measurement. Note that the basic video unit stored in Kankan is segment. Every segment is considered as an individual video object and assigned with a video ID. Thus, we collected the segment distribution of all the candidate movies and TV shows in Kankan. We deployed our CDN crawlers and P2P crawlers, and requested the same video segments to the Kankan RSs and TRs. The crawling experiments were conducted continuously for 24 hours on April 15, 2012. We recorded the replied CSs and peers. The measurement results are listed in Table VIII.

The video segment distribution is tabulated in Table VIII. Total, 1,122 movies and 36,774 TV shows are identified in our data set. Every movie is divided into two or more segments, while every TV show is kept as one segment. Examining the sample segments, we find that the average size of video segments is about 150 MB. In fact, the number of segments is affected by the coding algorithm and the content length. For example, the middle-quality TV episode is usually less than 45 minutes and thus forms one segment; the middle-quality movie consists of two segments; the high-quality movie may consists of up to 6 segments.

The macro-level content distribution can also be revealed in Table VIII. The movies and TV shows are found in all the harvested CDN CSs, but in different numbers of Kankan peers. About 1 million peers are harvested for the movie category, and about 2.9 millon peers for the TV show. After combining the two sets of peers and removing the repeated ones, we find unique 3,085,603 peers in total. Considering the periodical status exchange of Kankan peers, at least 3 million active Kankan clients are harvested in this daily measurement.

TABLE VIII
Video Contents in Kankan

Contents			Storage	
Category	Video	Segment	CDN CSs	Peers
Movie	1,122	2,454	258	1,008,486
TV show	36,774	36,774	258	2,876,652

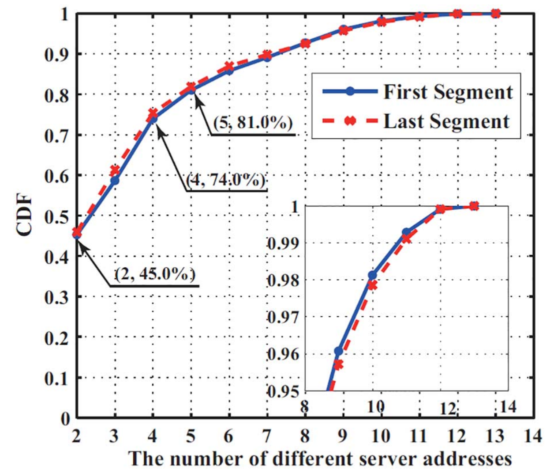


Fig. 8. CDF of content servers storing the first and the last movie segments.

C. Content Distribution in CDN and P2P Network

In this section, we further compare the distribution of the same video in the Kankan CDN and the P2P network. Meanwhile, we also aim to study the user behaviors that could be inferred from the video segments cached in peers. We focus on the movie category. Note that a movie video in Kankan usually consist of two or more segments. If the last segment of one movie is found in a peer cache, the corresponding user is expected to "watch" the video completely.

It is notable that there are some differences between the real user behaviors and the cached video distribution. The Kankan client prefetches future segments when a user playbacks video normally, but does not cache the skipped video in case of VCR operations. If a user watches the complete video or pauses for long enough time, the full length of video will be cached in the local peer cache. The cached content can be deleted manually by the users, or automatically replaced due to the caching policy. The cache occupance reported by the cache list is the aggregate result based on both user behaviors and the caching policy of the Kankan client. The cached video contents may be longer than the actual watched content. Despite the above discrepancy, the macro-level user behaviors can be inferred at least partially by tracking the cached videos in the Kankan peers.

We analyzed the cached contents for 2,454 movie segments in our data set, and plot the CDFs for the segment distribution in CDN servers and peers in Fig. 8 and Fig. 9, respectively.

In Fig. 8, the CDF curves of the first segment and the last segment stored in the Kankan CDN are almost identical, while

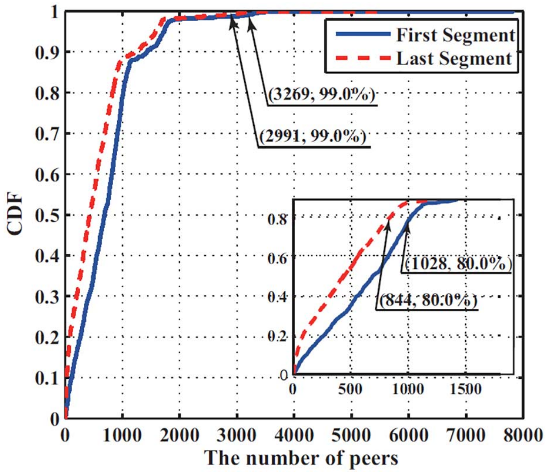


Fig. 9. CDF of peer caching for the first and the last movie segments.

the curve of the last segment is a bit higher. We conjecture that the different segments of the same movie are stored in the same content servers. The first segments are stored in more servers, and this is caused by the caching policies of Kankan SCS on the video prefix. Since the CDN may adjust the content distribution based on user requests, the number of CSs should be correlated with the video popularity. We then read Fig. 8 in terms of the popularity. The minimal number of CSs for each segment is 2. Thus, for any movie segment, at least 2 content servers are assigned for storage. Nearly 45.0% of the total videos are stored in only 2 CSs, which can be classified into the least visited videos. 74.0% least visited segments are stored in only 4 or less CSs, in other words, 26.0% top popular segments are stored in at least 5 CSs. The top 1.0% popular segments have more than 11 copies in the Kankan CDN.

In Fig. 9, the CDF curves of the segments in peer caches are quite different from those in the CDN. We also sort the segment distribution according to the video popularity. For the least visited movies, the number of caching peers is as low as zero. With an increasing popularity, the number of caching peers also increases, but the last segment is usually cached less than the first segment. For the 20.0% the most visited movies, the minimum number of peers caching the first and the last segment is 1,028 and 844, respectively. For the top 1.0% most visited movies, the number of peers caching the first and last segments increases to 3,269 and 2,991, respectively.

We compare the two popularity lists counted by the number of content servers and peers, and find the movies are cached following almost the same popularity order. For example, 91.0% of the top 1.0% most visited movies in the two lists are identical. Combining the results of Fig. 8 and Fig. 9, the top 1.0% most visited segments are stored in more than 11 CDN servers and 2,991 peers; 20.0% the most visited movies are stored in more than 5 CDN servers and 844 peers; the least visited segments are stored in only 2 CDN servers.

D. Content Dynamics in Peer Caches

In this section, we study the average duration of one video staying in the Kankan peer caches. Considering the content dynamics in peer caches, we adopt the number of peers as an indication of content activity. A metric, namely *content lifetime*, is used to analyze our data set. Since we are not able to track the Kankan P2P network for long duration beyond our measurement period, the content lifetime is upper bounded. We illustrate our definition of the content lifetime in Fig. 10.

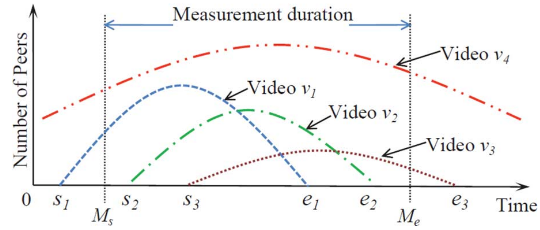


Fig. 10. Content lifetime in peer caches.

As shown in Fig. 10, we assume that the lifetime of one video has a hump-like shape in units of peers based on the video lifetime measurement in [28]. Due to the limited duration of our measurement, our emphasis is to estimate the shortest lifetime of the content cached in peers, rather than the longest one. As a result, we only consider the videos whose lifetime end in our measurement duration. We indicate the possible cases of lifetime in Fig. 10. Suppose that four videos are observed in the peer x, which are indicated as $v_i, (i = 1, \ldots, 4)$. Since the end points of v_3 and v_4 are beyond the measurement duration, only v_1 and v_2 will be included in our measurement. Note that, this measurement filtering rule excludes the videos with longest lifetime like v_4, and also underestimates the lifetime of the videos like v_1.

We denote the start and end time of video i as s_i and e_i ($s_i < e_i$), the start and end time of the measurement as M_s and M_e. Then, the content lifetime l_i of video i is defined as

$$l_i = \begin{cases} e_i - s_i, & \text{for} \quad s_i \geq M_s \quad \text{and} \quad e_i < M_e \quad (1) \\ e_i - M_s, & \text{for} \quad s_i < M_s \quad \text{and} \quad e_i < M_e \quad (2) \\ N/A, & \text{for} \quad e_i \geq M_e. \quad (3) \end{cases}$$

Our measurement duration lasted for one month from 13 April 2012 to 12 May 2012. We took the movie category as target, and crawled the Kankan P2P network in every hour. Total 1,269 movies were observed, in which 35 were found in the case of v_1, 4 of v_2, 173 of v_3 and 1,057 of v_4. It is notable that, a major proportion (83.3%) of the movies are in the case of v_4, having the lifetime longer than the measurement duration. To better understand the cache dynamics, we focus on the shortest lifetime cases (v_1 and v_2), and select the 39 movies in our further study.

We searched the records of peer lists, and tracked 98,950 distinct peers caching these 39 movies. We sort the movies according to the maximum number of caching peers. The lifetime of No. 1, No. 15 and No. 30 movies are plotted in Fig. 11. Considering the dynamics of P2P network, the peers appeared in the 1st day may be much different to those appeared in the 30th day. It is obvious that, the in-cache lifetime of these three movies have the similar shapes as we expected in Fig. 10. Therefore, the content lifetime defined in this paper is practical and can be measured in real traces.

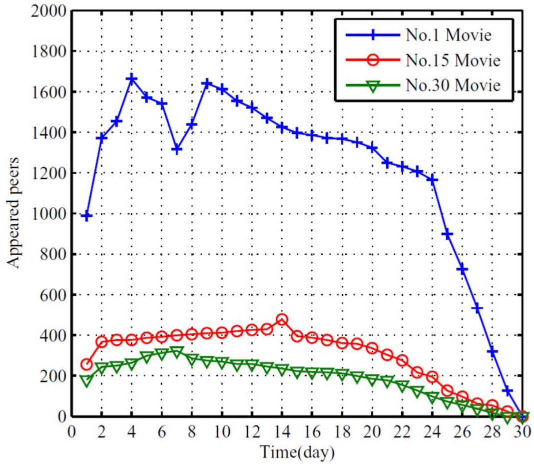


Fig. 11. Measured lifetime of movies in peer caches.

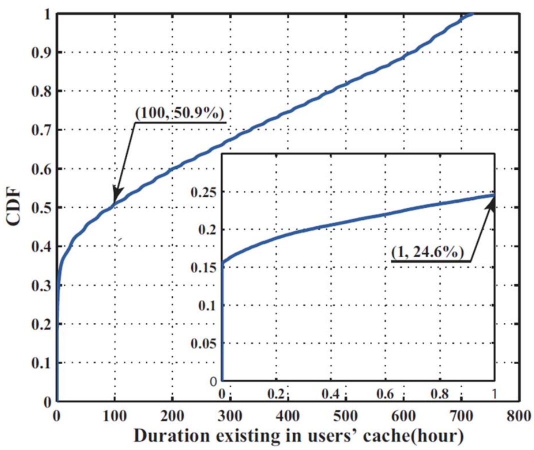


Fig. 12. CDF of video lifetime in peer cache.

One specific movie may "live" different durations in the viewpoints of different peers. We collected all these views of content lifetime, which are more than one hundred thousand records, and plot the CDF of in-cache lifetime distribution of the monitored movies in Fig. 12. The average lifetime of the cached videos is 209.5 hours, which is longer than one week. There also exist some videos with quite short lifetime. In 24.6% of the total records, the lifetime of cached contents are less than one hour. On the other hand, in nearly half (49.1%) of the total records, the lifetime of cached videos is more than 100 hours.

Note that our measurement method in Fig. 11 imposes a truncation effect on the measurement results, which results in shorter lifetime than the real case. Considering this under-estimation, we conclude that the content distribution in Kankan peers changes quite slowly. This provides a significant advantage of the Kankan P2P network in content sharing.

E. Summary

We summarize the major measurement results on the content distribution in Kankan as follows.

- *Scale of the Kankan P2P network*: The Kankan P2P network consists of a large number of peers. About 1 million peers were found for the movie category in one day, and about 2.8 million peers for the TV shows. At least 3 million active Kankan clients were observed just one day.
- *Video distribution in the Kankan CDN and P2P network*: The Kankan CDN is responsible for storing all the videos, while the peers cache the video segments based on user behaviors. 20.0% most popular movie segments are stored in at least 5 CDN servers and 844 peers. Most unpopular movie segments are only stored in 2 CDN servers.
- *Lifetime of videos cached in the Kankan peers*: The local cache of Kankan peers is determined by user behaviors and the Kankan client caching polices. The video content cached in Kankan peers updates quite slowly. In our measurement data set, the average lifetime of the cached movies is 209.5 hours. 49.1% of them have more than 100 hours lifetime in peer caches.

VII. Conclusion

In this paper, we conducted comprehensive active and passive measurements on Xunlei Kankan, the leading hybrid CDN-P2P VoD streaming provider in China. We performed an in-depth protocol analysis on Kankan's encrypted proprietary protocol and uncover a large portion of its signaling messages. We unreeled the basic CDN-P2P architecture of the Xunlei Kankan VoD streaming system, and dissected the switching mechanisms between CDN and P2P in the Kankan VoD Streaming. We also characterized the dynamics of video contents cached in the Kankan P2P network.

We find two important technical features of Kankan system:

1) *Loosely-coupled CDN-P2P architecture*: Kankan adopts a loosely-coupled hybrid CDN-P2P architecture, in which the CDN and the P2P network are requested by Kankan clients separately. The major reason of such a design could be the large-scale content storage in Kankan peers. Inferred from our measurements, the average daily active Kankan clients are more than three million, and the contents cached in Kankan peers change quite slowly. Taking the advantages of the P2P storage, Kankan efficiently reduces the number of required CDN servers. The total system operation cost of Kankan will be much smaller than the known CDN-based VoD providers.

2) *CDN-assisted hybrid streaming*: The Kankan CDN is deployed in the close proximity to the geographic areas with large netizens or clients in China. Kankan deploys some redirection servers to dispatch the requests to CDN content servers. The dual-server streaming enhancement is only adopted in the startup phase, and the client only requests CDN assistance when the P2P streaming capacity is insufficient. It is notable that, besides the effectively utilizing P2P capacity, Kankan achieves good system scalability as well. The adoption of specific redirection servers can be easily extended to support a larger-scale of CDN. Multiple server streaming could also be easily implemented in the current framework.

The lifetime measured in this paper could be used to evaluate the efficiency and performance of VoD streaming systems. In future work, we would like to utilize these measurement traces and perform trace-driven simulation studies on the hybrid streaming.

REFERENCES

[1] iResearch Inc. Shanghai, China, "iUserTracker: Rank of video and audio player software in China (2012.12) (in Chinese)," [Online]. Available: http://service.iresearch.cn/software/20130206/192729.shtml , accessed Feb. 2013

[2] P. Dhungel, K. W. Ross, M. Steiner, Y. Tian, and X. Hei, "Xunlei: Peer-assisted download acceleration on a massive scale," in *Proc. PAM*, 2012, pp. 231–241.

[3] D. Xu, S. S. Kulkarni, C. Rosenberg, and H.-K. Chai, "Analysis of a CDN-P2P hybrid architecture for cost-effective streaming media distribution," *Multimedia Syst.*, vol. 11, no. 4, pp. 383–399, Apr. 2006.

[4] C. Huang, A. Wang, J. Li, and K. W. Ross, "Understanding hybrid CDN-P2P: Why limelight needs its own red swoosh," in *Proc. ACM NOSSDAV*, 2008, pp. 75–80.

[5] C. Hu, M. Chen, C. Xing, and B. Xu, "Eue principle of resource scheduling for live streaming systems underlying CDN-P2P hybrid architecture," *Peer-to-Peer Netw. App.*, vol. 5, no. 4, pp. 312–322, 2012.

[6] E. Baccaglini, "A study of an hybrid CDN-P2P system over the planetlab network," *Signal Process.: Image Commun.*, vol. 27, no. 5, pp. 430–437, May. 2012.

[7] A. Balachandran, V. Sekar, A. Akella, and S. Seshan, "Analyzing the potential benefits of CDN augmentation strategies for Internet video workloads," in *Proc. ACM IMC*, 2013, pp. 43–56.

[8] H. Yin *et al.*, "LiveSky: Enhancing CDN with P2P," *ACM Trans. Mutimedia Comput., Commun., Appl.*, vol. 6, no. 3, pp. 16–34, Aug. 2010.

[9] M. Zhao *et al.*, "Peer-assisted content distribution in Akamai netsession," in *Proc. ACM IMC*, 2013, pp. 31–42.

[10] Z. Lu, Y. Wang, and Y. R. Yang, "An analysis and comparison of CDN-P2P hybrid content delivery system and model," *J. Commun.*, vol. 7, no. 3, pp. 232–245, Mar. 2012.

[11] V. K. Adhikari, S. Jain, and Z.-L. Zhang, "YouTube traffic dynamics and its interplay with a tier-1 ISP: An ISP perspective," in *Proc. ACM IMC*, 2010, pp. 431–443.

[12] V. K. Adhikari, S. Jain, Y. Chen, and Z.-L. Zhang, "Vivisecting YouTube: An active measurement study," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2521–2525 .

[13] V. K. Adhikari, "Unreeling Netflix: Understanding and improving multi-CDN movie delivery," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1620–1628.

[14] V. K. Adhikari, "A tale of three CDNs: An active measurement study of Hulu and its CDNs," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Mar. 2012, pp. 7–12.

[15] Y. Huang, "Challenges, design and analysis of a large-scale P2P-VoD system," in *Proc. ACM SIGCOMM*, 2008, pp. 375–388.

[16] J. Lei, L. Shi, and X. Fu, "An experimental analysis of Joost peer-to-peer VoD service," *Peer-to-Peer Netw. Appl.*, vol. 3, no. 4, pp. 351–362, 2010.

[17] Y. Tian, R. Dey, Y. Liu, and K. W. Ross, "Topology mapping and geolocating for China's internet," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 9, pp. 1908–1917, Sep. 2013.

[18] X. Liu, "A case for a coordinated Internet video control plane," in *Proc. ACM SIGCOMM*, 2012, pp. 359–370.

[19] A. Mansy and M. Ammar, "Analysis of adaptive streaming for hybrid CDN/P2P live video systems," in *Proc. IEEE ICNP*, Oct. 2011, pp. 276–285.

[20] A. Arefin, Z. Huang, K. Nahrstedt, and P. Agarwal, "4D TeleCast: Towards large scale multi-site and multi-view dissemination of 3DTI contents," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, Jun. 2012, pp. 82–91.

[21] Netflix. Los Gatos, CA, USA, "Open connect content delivery network," [Online]. Available: https://www.netflix.com/openconnect , accessed Jun. 2012

[22] Y. Chen, B. Zhang, Y. Liu, and W. Zhu, "Measurement and modeling of video watching time in a large-scale internet video-on-demand system," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 2087–2098, Dec. 2013.

[23] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao, "YouTube everywhere: Impact of device and infrastructure synergies on user experience," in *Proc. ACM IMC*, 2011, pp. 345–360.

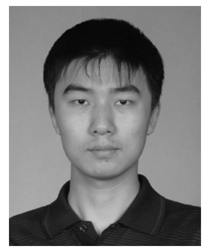
[24] iResearch Inc. Shanghai, China, "iUserTracker: Rank of download software in China (2014.3) (in Chinese)," [Online]. Available: http://service.iresearch.cn/others/20140430/230555.shtml , accessed Apr. 2014

[25] iResearch Inc. Shanghai, China, "PPTV ranked no.1 by daily user traffic in China online video industry," [Online]. Available: http://www.iresearchchina.com/news/4262.html , accessed Jun. 2012

[26] C. Huang, A. Wang, J. Li, and K. W. Ross, "Measuring and evaluating large-scale CDNs," in *Proc. ACM IMC*, 2008, pp. 15–29.

[27] China Internet Netw. Inf. Center, Beijing, China, "Statistical report on internet development in China," Jan. 2013 [Online]. Available: http://www1.cnnic.cn/IDR/ReportDownloads/201302/P020130312536825920279.pdf

[28] H. Abrahamsson and M. Nordmark, "Program popularity and viewer behaviour in a large TV-on-demand system," in *Proc. ACM IMC*, 2012, pp. 199–210.



Ge Zhang received the B.S. and M.S. degrees in computer science from the Zhongnan University of Economics and Law, Wuhan, China, in 2005 and 2007, respectively, and is currently working toward the Ph.D. degree at the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China.

His research interests include streaming media, network modeling, and measurement.

Wei Liu (M'07) received the B.S. degree in telecommunication engineering and Ph.D. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2004, respectively.

He is currently an Associate Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology. His research interests include Internet streaming and measurement and content-centric networks.

Xiaojun Hei (S'02–M'07) received the B. Eng. degree from the Huazhong University of Science and Technology, Wuhan, China, in 1998, and the M. Phil. and Ph.D. degrees from the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, in 2000 and 2008, respectively.

Since 2008, he has been an Associate Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology. His current research interests include network measurement and software-defined network.

Wenqing Cheng (M'07) received the B.S. degree in telecommunication engineering and Ph.D. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1985 and 2005, respectively.

She is currently a Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology. Her research interests include information systems and e-learning applications.