Insights into PPLive: A Measurement Study of a Large-Scale P2P IPTV System [*]

Xiaojun Hei[†], Chao Liang[‡], Jian Liang[†], Yong Liu[‡] and Keith W. Ross[†]
[†]Department of Computer and Information Science
[‡]Department of Electrical and Computer Engineering
Polytechnic University, Brooklyn, NY, USA 11201

ABSTRACT

With over 100,000 simultaneous users (typically), PPLive is the most popular IPTV application today. PPLive uses a peer-to-peer design, in which peers download and redistribute live television content from and to other peers. Although PPLive is paving the way for an important new class of bandwidth intensive applications, little is known about it due to the proprietary nature of its protocol. In this paper we undertake a preliminary measurement study of PPLive, reporting results from passive packet sniffing of residential and campus peers. We report results for streaming performance, workload characteristics, and overlay properties.

General Terms

Measurement, Performance

Keywords

IPTV, Peer-to-Peer Measurement, PPLive

1. INTRODUCTION

IPTV is expected to be the next disruptive IP communication technology, potentially reshaping our media and entertainment culture [1]. However, provisioning the IPTV service brings forth significant new challenges [2]. IPTV systems can be broadly classified into two categories: infrastructure-based and peer-to-peer based. In infrastructure-based systems, video servers and application-level multicast nodes are strategically placed in the Internet, and video is streamed from servers to clients via the multicast nodes. The infrastructure-based IPTV systems are expensive to build and difficult to maintain. On the other hand, peer-to-peer IPTV systems do not rely on dedicated application-level multicast servers. Instead, each IPTV client is potentially a server, multicasting received content to other IPTV clients. The IPTV clients and the connections between them thus form an overlay network, cooperatively exchanging video content by leveraging the uploading capacity of the peers.

Several P2P IPTV systems have been developed and deployed to date. Among them, CoolStreaming [3] and PPLive [4] have been two of the most popular P2P-based IPTV applications. Zhang et al. [3] reported that more than $4,000$ CoolStreaming users were simultaneously online at some peak time. Using a crawler (to be discussed in a subsequent paper), we observed in our PPLive measurements more than $100,000$ simultaneously online users for a live broadcast of a popular TV program. To the best of our knowledge, PPLive is by far the most popular P2P IPTV application on the Internet today. Given its success and its low-cost P2P architecture, it is our position that the designers of future IPTV systems should understand PPLive's design, performance, traffic characteristics and, more broadly, its strengths and its flaws. Nevertheless, PPLive employs proprietary signaling and video delivery protocols. Details about its performance, streaming workload and overlay characteristics are still largely unknown.

In this paper we present results from a preliminary measurement study of PPLive. We have been measuring PPLive with passive packet sniffing as well as with an active crawler. In this paper, we only present the sniffing results. Our measurement study focuses on three important aspects of PPLive streaming: streaming performance, workload characteristics, and overlay properties. Quantitative results obtained in our study bring to light important performance and design issues of live streaming over the public Internet.

2. OVERVIEW OF PPLIVE

PPLive is a free P2P-based IPTV application. According to the PPLive web site [4] in January 2006, the PPLive network provides $200+$ channels with $400,000$ daily users on average. The bit rates of video programs mainly range from 250 Kbps to 400 Kbps with a few channels as high as 800 Kbps. The PPLive network does not own video content. The video content is mostly feeds from TV channels in Mandarin. The channels are encoded in two video formats: Window Media Video (WMV) or Real Video (RMVB). The encoded video content is divided into chunks and distributed to users through the PPLive P2P network. The PPLive web site [4] provides limited information about its video content distribution mechanism. Nevertheless, various web sites and message boards provide additional information. In this section we describe some of PPLive's fundamental mechanisms, which we collected from different sources and confirmed by our own measurement results.

The PPLive software implements two major application level protocols: a gossip-based protocol for peer management and channel discovery; and a P2P-based video distribution protocol for high quality video streaming. Figure 1 depicts an overview of the PPLive network. When an end-user starts the PPLive software, it joins the

---

[*] This work is supported in part by NSF under grant ITR-0325726 and grant CNS0519998. Any opinions, findings, and conclusions of the authors do not necessarily reflect the views of NSF.

IPTV Workshop, International World Wide Web Conference May 23, 2006, Edinburgh, Scotland, United Kingdom

PPLive network and becomes a PPLive peer node. It then sends out a query message to the PPLive channel server to obtain an updated channel list (step 1). Before a peer actually starts to watch a channel, it does not exchange data with other PPLive peers. After a peer selects one channel to watch, it sends out multiple query messages (step 2) to some root servers to retrieve an online peer list for this channel. Peers are identified by their IP addresses and port numbers on the list. Upon receiving a peer list, the PPLive client sends out probes to peers on the list to find active peers for the channel of interest (step 3). Some active peers may also return their own peer lists, helping the initial peer to find more peers. Peers then share video chunks with each other, as described below.

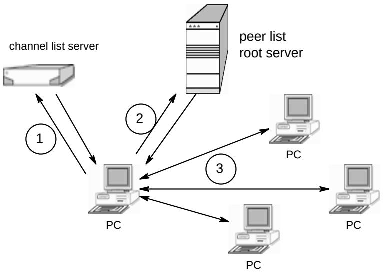


Figure 1: Channel and peer discovery

The major software component of PPLive is its TV engine. This TV engine is responsible for downloading video chunks from the PPLive network and streaming the downloaded video to a local media player. The streaming process in the PPLive traverses two buffers in local memory: the PPLive TV engine buffer and the media player buffer, as shown in Figure 2. This double buffering mechanism is designed with two goals. One is to pre-cache media content to combat download rate variations from the PPLive network; the other is for efficient content distribution between peers.

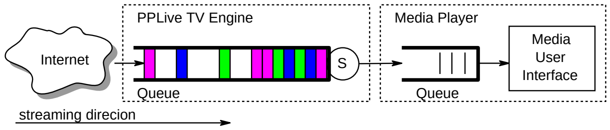


Figure 2: PPLive streaming process

The cached contents can be uploaded to other peers that are watching the same channel. Specifically, the peer client contacts multiple active peers to download media content of the channel. At the same time, this peer client may also upload cached video chunks to multiple peers. Received video chunks are reassembled in order and buffered in the queue of the PPLive TV engine, forming a local streaming file in memory.

When the streaming file length crosses a predefined threshold, the PPLive TV engine launches a media player, which downloads video content from the local HTTP streaming server. Most media players, such as windows media player, have built-in video buffering mechanisms. After the buffer of the media player fills up to the required level, the actual video playback starts.

When PPLive starts, the PPLive TV engine downloads media content from peers aggressively to minimize the playback start-up delay. When the media player receives enough content and starts to play the media, the streaming process gradually stabilizes. The PPLive TV engine streams data to the media player at the media playback rate.

3. MEASUREMENT SETTING

Our P2P network measurements fall into two categories: passive monitoring and active crawling. In this paper, we describe the results from the passive monitoring platform, which captures PPLive traffic. We collected multiple PPLive packet traces from four PCs: two PCs were connected to Polytechnic University campus network with 100 Mbps Ethernet access; two PCs were connected to residential networks through cable modem. Most of the PPLive users today have either one of these two types of network connections. The PCs with residential access were located at Manhattan and Brooklyn in New York. Each PC ran Ethereal [5] to capture all inbound and outbound PPLive traffic. We carefully filtered out the cross traffic of other network activity from the PCs.

Table 1 provides an overview of the collected packet traces, which were captured on February 2, 2006. One residential and one campus PC "watched" the channel CCTV3; the other residential and campus PC "watched" the channel CCTV10. Each of these four traces lasted about 2 hours. From the PPLive web site, CCTV3 is a popular channel with a 5-star popularity grade and CCTV10 is a less popular channel with a 3-star popularity grade. We counted an IP address in these traces if there was a TCP connection attempt (TCP SYN packets) between the traced peer and this IP address. We define an IP address to be active if the peer with this IP address exchanges non-zero data with the traced peer.

4. STATISTICS OF PPLIVE SESSIONS

During playback, a PPLive peer normally establishes a large number of sessions to other peers, not only for content exchange but also for signaling. In this section, we present detailed session statistics, such as session duration, packet size and the correlation between them, and traffic breakdown among sessions.

4.1 Session Duration and Packet Size

A PPLive client utilizes TCP for both signaling and video streaming. TCP signaling sessions normally perform short-duration tasks, including downloading peer lists and probing peers for availability. TCP streaming sessions, on the other hand, have longer durations. Furthermore, TCP streaming packets normally has a large packet size, 1200+ bytes, while small TCP signaling packets are commonly observed. We plot the correlation between TCP session durations and average TCP segment size in Figure 3 for CCTV3-Campus. The plots for the other three traces are similar. We clearly observe that long TCP flows mainly have large TCP segment sizes. There exists many TCP sessions with short durations and small TCP segment sizes. From Figure 3, we can conclude that signaling sessions typically have short durations and carry mostly small packets; whereas video exchange sessions have long durations and carry many large packets.

The presence of signaling and streaming traffic makes it difficult to understand the PPLive working mechanisms. Hence, we separate video and signaling traffic with the aid of a heuristic:

1. For a given TCP session, we keep track of the cumulative number of large packets ($>$ 1200 bytes) during a session's lifetime. If the cumulative number of large packets is larger than 10, this session is labelled as a "video session"; otherwise, the session is labelled as a "signaling session". We filter from the traces all signaling sessions.
2. Within a video session, we further filter all packets less than 1200 bytes.

We also provide of a typical Complementary Cumulative Distribution Function (CCDF) of video session durations in Figure 4.

Table 1: Data sets

Trace Name	Trace size (Byte)	Duration (Sec)	Playback Rate (Kbps)	Total IPs	Active IPs	Download (MByte)	Upload (MByte)
CCTV3-Campus	784,411,647	7676	340	3105	2691	360.99	4574.57
CCTV3-Residence	132,494,879	7257	340	1616	1183	372.53	352.75
CCTV10-Campus	652,000,813	7285	312	1008	910	317.08	3815.34
CCTV10-Residence	66,496,909	9216	312	797	282	385.50	7.68

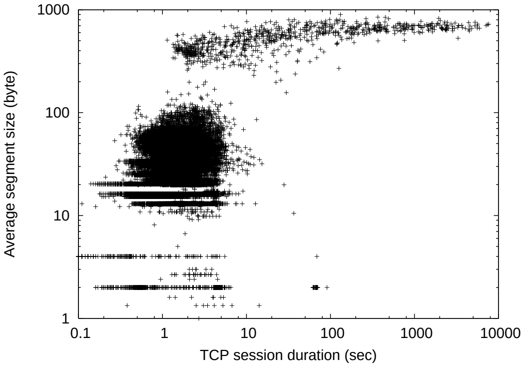


Figure 3: TCP session duration vs. TCP average segment size for CCTV3-Campus

Note that the video session duration spreads over a wide range. The median video session is about 20 seconds and about 10% of video sessions last for over 15 minutes or more. Because many sessions are short, a peer may only exchange a few video chunks with its neighbors before the session ends.

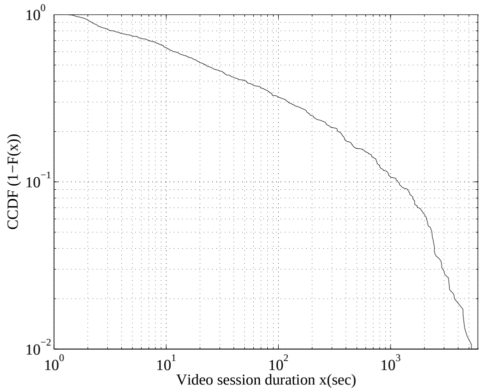
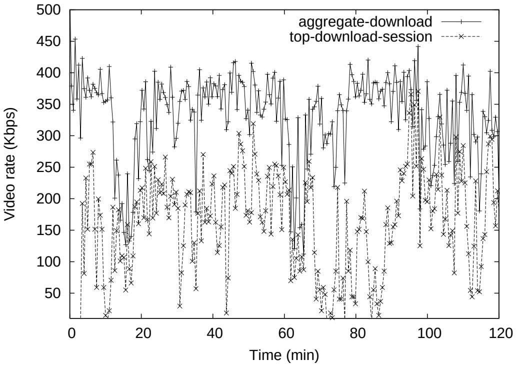


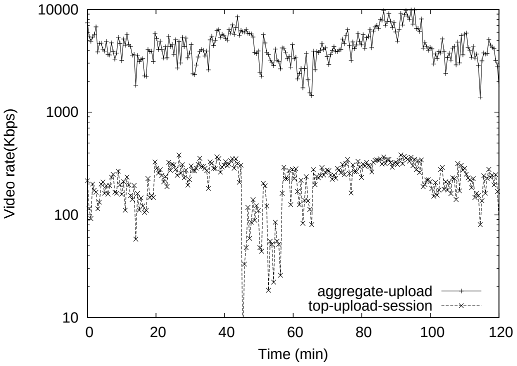
Figure 4: CCDF of video session duration for CCTV3-Campus

4.2 Video Traffic Breakdown among Sessions

A PPLive peer downloads/uploads video chunks from/to multiple peers. However, due to bandwidth diversity and content availability on peers, download/upload rates of all peering sessions are not evenly distributed. In Figure 5(a), for a campus peer, we compare its aggregate download video rate with the download rate from the greatest contributing peer. This top peer almost contributes about 50% of the total video download traffic. However, the download rate from this top peer is highly dynamic. This is mostly due to the content availability on the top peer and the inherent rate variation of the TCP session with that peer. One important consequence is that a peer typically receives video from more than one peers at any given time. With this multi-download feature, the aggregate video download rate becomes quite smooth. In conjunction with the buffering mechanisms, which will be discussed in Section 5.4, PPLive is typically able to provide smooth video playback. We also plot analogous curves, in log scale, for video upload in Figure 5(b). Importantly, the top peer video upload session only takes account for about 5% of the total video upload traffic. Thus this campus node is performing an important multicast function.



(a) Download



(b) Upload

Figure 5: Peer download and upload video traffic breakdown for CCTV3-Campus

5. PPLIVE STREAMING PERFORMANCE

5.1 Start-up Delay

When PPLive first starts, it requires some time to search for peers and then tries to download data from active peers. We record two types of start-up delay: the delay from when one channel is selected until the streaming player pops up; and the delay from when the player pops up until the playback actually starts. The player pop-up delay is in general $10 \sim 15$ seconds and the player buffering delay is around $10 \sim 15$ seconds. Therefore, the total start-up delay is around $20 \sim 30$ seconds. Nevertheless, we observe that some less popular channels have a total start-up delays of up to 2 minutes. Overall, PPLive exhibits reasonably good start-up user experiences, which is confirmed by the quick ramp-up of the download traffic rate shown in Figure 6.

5.2 Upload-Download Rates

Figure 6 depicts the upload and download streaming rates for the four traces. Each data point is the average bit rate over 30 second intervals. Note that the download bit rates quickly ramp up to the playback rate. It is also interesting that the upload traffic rates increase very quickly in the two campus traces; however, the upload rate of CCTV3-Residence increases very slowly (see Figure 6(b)) and there is essentially no upload traffic from CCTV10-Residence.

We observe that in general, the video download rates for both campus peers and residential peers smoothly fluctuate around their video playback rates. However, there is an obvious download rate decrease for the residential peer in trace CCTV10-Residence around time $t = 33$ minute. This rate fluctuation period sustains for around 4 minutes. After this decrease, the PPLive TV engine aggressively downloads from the network and speeds up the download rate for another 3 minutes. Then the download rate becomes steady again. Despite of the PPLive and media player buffering, this download rate fluctuation may have impacted the quality of video playback.

For upload traffic rates, campus peers and residential peers exhibit distinctly different behaviors. The two campus peers upload significantly more traffic than the two residential peers. Over the 2-hour period, the two campus peers uploaded about 4.4 GBytes and 3.7 GBytes video traffic to other peers, respectively. Although not as high as the two campus peers, one of the residential peers contributed traffic volume comparable to its download traffic volume. However, the other residual peer only uploaded 4.6 MBytes video chunks to other peers.

5.3 Video Traffic Redundancy

Due to the distributed nature of PPLive streaming, it is possible that a PPLive peer downloads duplicate media content from multiple peers. The transmission of redundant video chunks wastes network bandwidth; hence, we are interested in the redundancy measurement of the PPLive video traffic after the streaming player playbacks steadily. To this end, the first 10 minutes of the traces are not used for analysis to minimize the impact of transient behavior of the traces. Excluding TCP/IP headers, we determine the total streaming payload for the download traffic. Utilizing the video traffic filtering heuristic rule, presented in Section 4, we are able to extract video traffic. Given the playback interval and the media playback speed, we obtain a rough estimate of the media segment size. We compute the redundant traffic by the difference between the total received video traffic and the estimated media segment size. We define the redundancy ratio as the ratio between the redundant traffic and the estimated media segment size. From Table 2, we observe that the traffic redundancy in PPLive is limited. This is partially due to the long buffer time period so that PPLive peers have enough time to locate peers in the same streaming channel and exchange content availability information between themselves.

The negative redundancy ratio (-3.5%) for CCTV3-Campus indicates that the video download chunks are not sufficient for smooth video playback. As shown in Figure 6(a), at time $10 < t < 20$ minute and $60 < t < 64$ minute for CCTV3-Campus, the download rate decreases significantly and the PPLive playback may suffer seriously lacking of video chunks. Given the good connectivity of campus network, this abnormal case requires further investigation.

5.4 Video Buffering

During periods of network congestion and peer churn, the media download rate may not sustain the normal media playback rate, causing the playback buffer to drain. Therefore, the buffer size affects the streaming application's resilience to network congestion. We estimate the buffer size of both the PPLive TV engine and the media player in the rest of this section.

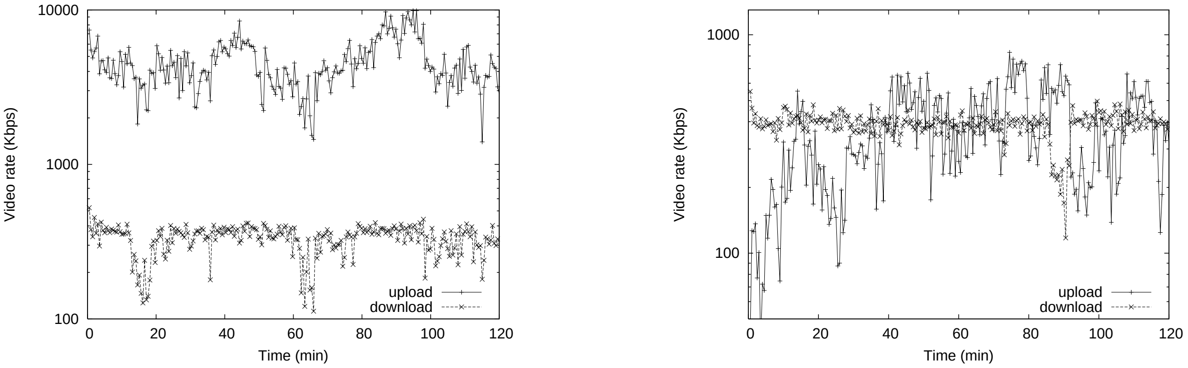
We estimate the media player buffer as follows. We first start to play one streaming channel and wait until the player begins to play. The media playback rate, c, can be read from the media player interface. After a time period, the speed and peer number displayed on the PPLive TV engine become stabilized. We then close the PPLive TV engine at time instance t_1. The media player continues playback the video chunks in its own buffer. Finally, the player reports the end of the program at time instance t_2. We calculate the time interval $(t_2 - t_1)$ and multiply it with the playback rate c to estimate the buffer size of the media player. Note that after we shut down the PPLive TV engine, the data already stored in the PPLive queue are no longer available for the media player. Therefore, the media player buffer is at least $c(t_2 - t_1)$. Multiple experiments indicate that this buffer size is at least 5.37 MBytes.

We estimate the buffer size of the PPLive TV engine as follows. First, we play one streaming channel to reach a steady streaming state. We physically disconnect the PC from the network. At the same time, we launch an HTTP file download software to download the media file from the PPLive streaming server. Note that after the network cable is unplugged, the PPLive TV engine still serves as a streaming server. The size of the downloaded video file is a rough estimate of the PPLive buffer size. Over multiple experiments for different streaming channels with variable rates, the estimated PPLive buffer size varies from 7.8 MBytes to 17.1 MBytes. It appears that the PPLive adaptively allocates buffer size according to the streaming rate and the buffering time period specified by the media source. Overall, the total buffer size in PPLive streaming $10 \sim 30$ MBytes. A commodity PC can easily meet this buffer requirement.

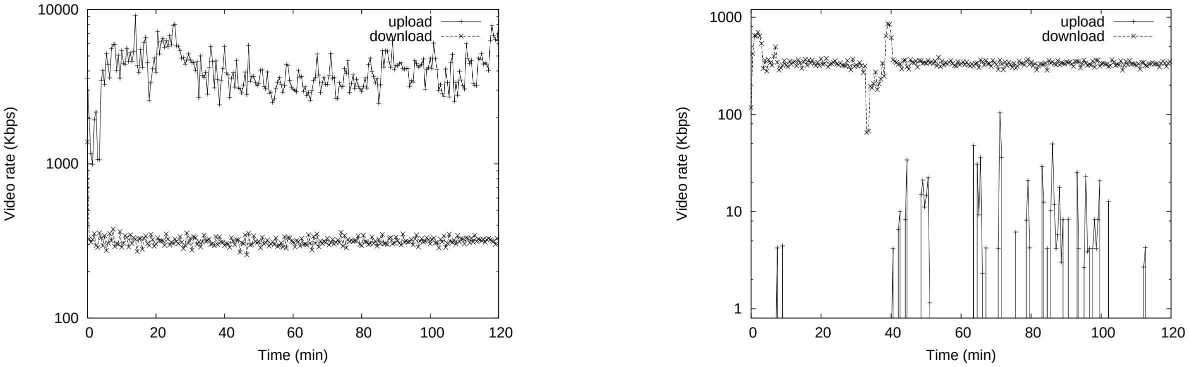
6. PPLIVE PEERING STATISTICS

Figure 6 plots the number of active video peers. Active video peers are defined as those peers which have more than 10 large packet (> 1200 bytes) exchange with the traced peer in its lifetime. There is distinct peer connectivity behavior for campus peers and for residential peers. As expected, a campus peer has many more active video peer neighbors than a residential peer due to its high-bandwidth access network. A campus peer utilizes its high-bandwidth connectivity, maintaining a steady number of active TCP connections for video traffic exchange. It also appears that content popularity has a significant impact on the number of active peer neighbors for the residential peer. In particular, the residential peer with the less popular CCTV10 channel seems to have difficulty in finding enough peers for streaming the media. At time $t = 33$ minutes, the active video peer number drops to 1. This reduction in video neighbors impacts the download rate of this residential peer significantly, as shown in Figure 6(d). In this experiment, the PPLive client detected this rate reduction quickly and started to search for new peers for additional video download. New peers were quickly found and fresh streaming flows were established; hence, the video download rate recovered quickly as a result.

During a peer's lifetime, this peer constantly changes its upload and download neighbors. This is illustrated in Figure 6, in which



(a) CCTV3-Campus (b) CCTV3-Residence



(c) CCTV10-Campus (d) CCTV10-Residence

Figure 6: Upload and download video bit rates for the four traces

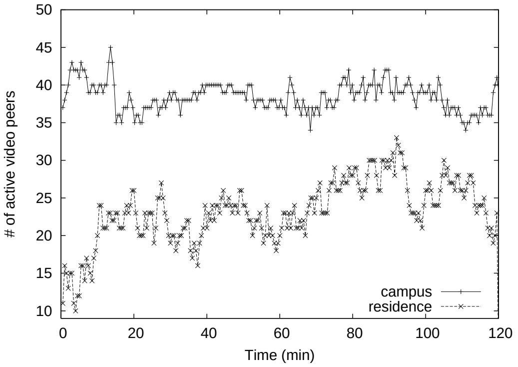
Table 2: Video traffic redundancy

Trace name	Interval (second)	Total download (MByte)	Video download (MByte)	Estimated media segment size (MByte)	Redundancy ratio
CCTV3-Campus	6966.2	308.3	285.7	296.1	-3.5%
CCTV3-Residence	6512.6	338.4	314.9	276.8	13.8%
CCTV10-Campus	6600.7	281.0	259.4	257.4	0.76%
CCTV10-Residence	8230.5	375.5	351.6	321.0	9.5%

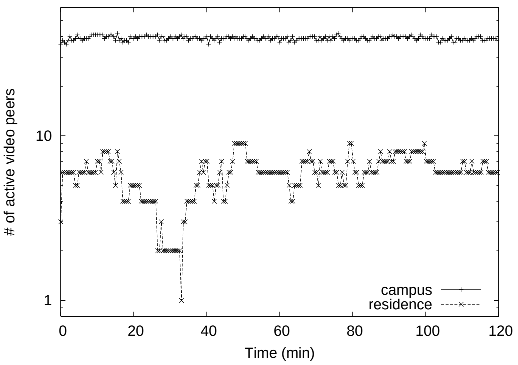
the number of video peers is sampled every 30 seconds. A changed video peer between two consecutive sampling time points (30 seconds) refers to a peer that either stops to serve as a video peer or becomes a new video peer for the traced peer. Regardless the types of access networks, over 30 seconds period we commonly observe that several video peers are gone and several new video peers start to exchange video chunks with the traced peer. Nevertheless, compared with the total number of video peers, the average number of the changed peers is less than 10% of the total video peers for campus peers. However, the changed peers contribute a large percentage of the total video peers for residential peers. One consequence is that the download video rates of residential peers are likely to fluctuate more significantly.

It would waste network resources to download from another continent if a channel can be downloaded from a source in the same continent. We investigated whether a PPLive peer takes locality into account when it determines which peer to download from. To this end, we employed a simple prefix matching technique to determine the geographic location of a peer. The first prefix byte of a peer's IP address is selected to estimate the geographic distribution of this peer. For example, 58.a.b.c is regarded as a peer from Asia. Note that there is still a small possibility that two IP addresses with the same prefix are located in different continents.

Table 3 shows the peer geographic distribution of IP addresses for the video sessions from the traces. We observe that a large number of peers are located in Asia and they contribute the majority of the download traffic for the traced peers as shown in Table 3. On the other hand, the majority of the video traffic uploaded by our traced peers, located in New York, is to peers in North America. For example, in Table 3(b), this residential peer downloads 81.9% video traffic from peers in Asia and 17.8% video traffic from peers in North America; however, it uploads only 5.4% video traffic to Asia but 64.8% to peers in North America. Nevertheless, Table 3(d) shows that this trend seems not to be valid for trace CCTV10-Residence. A closer investigation on this trace reveals that this residential peer uploaded few video chunks to a limited number of peers (see Figure 6(d)). Those video sessions are short-lived and those peers are only transient peers, largely distributed over the



(a) CCTV3



(b) CCTV10

Figure 7: Evolution of active video peer connections

global Internet.

7. CONCLUSION

We conducted a measurement study on a popular IPTV application, PPLive. Our measurement results show that the PPLive deploys the P2P principles for efficient resource discovery and video distribution. Utilizing the best-effort Internet infrastructure, the PPLive streaming maintains satisfactory IPTV performance. This demonstrates that the current Internet infrastructure is capable to provide economic-viable IPTV services while meeting the performance requirements of IPTV. Nevertheless, the emerging IPTV applications exhibit different characteristics from other applications, which may change Internet traffic pattern significantly. This brings forth new challenges and opportunities for network service providers.

8. REFERENCES

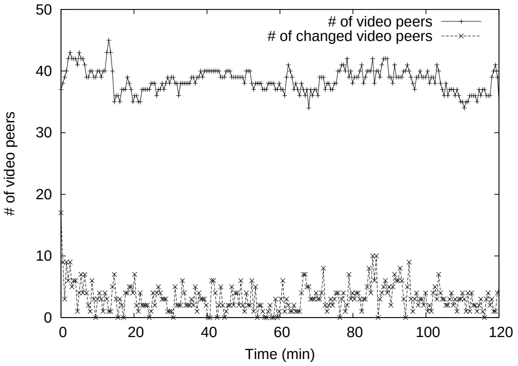
[1] S. Cherry, “The battle for broadband [Internet protocol television],” *IEEE Spectrum*, vol. 42, no. 1, pp. 24 – 29, Jan. 2005.

[2] R. Jain, “I want my IPTV,” *IEEE MultiMedia*, vol. 12, no. 3, p. 96, 2005.

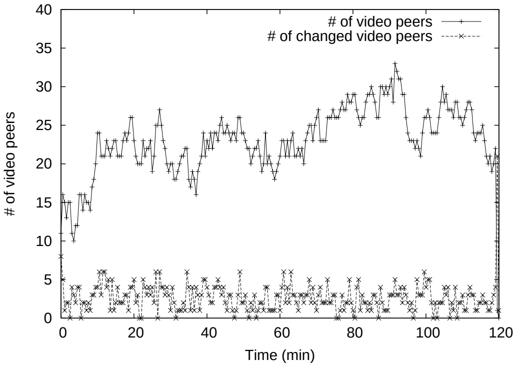
[3] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, “DONet/CoolStreaming: A data-driven overlay network for peer-to-peer live media streaming,” in *Proceedings of INFOCOM*, vol. 3, Miami, FL, USA, 13-17 March 2005, pp. 2102 – 2111.

[4] “PPLive.” [Online]. Available: http://www.pplive.com

[5] “Ethereal.” [Online]. Available: http://www.ethereal.com/



(a) CCTV3-Campus



(b) CCTV3-Residence

Figure 8: Peer departures and arrivals

Table 3: Peer geographic distribution of video sessions

(a) CCTV3-Campus

	Asia	North America	Other Places
peer(%)	18.6	73.0	8.4
Download(%)	77.3	21.6	1.1
Upload(%)	1.1	83.0	15.9

(b) CCTV3-Residence

	Asia	North America	Other Places
peer(%)	64.9	28.4	6.7
Download(%)	81.9	17.8	0.3
Upload(%)	5.4	64.8	29.8

(c) CCTV10-Campus

	Asia	North America	Other Places
peer(%)	36.1	55.3	8.6
Download(%)	94.6	4.9	0.4
Upload(%)	2.6	75.8	21.6

(d) CCTV10-Residence

	Asia	North America	Other Places
peer(%)	60.3	35.6	4.1
Download(%)	48.1	50.4	1.5
Upload(%)	45.7	24.8	29.5