2011 Ninth Annual Communication Networks and Services Research Conference

Measurement Study on PPLive Based on Channel Popularity

Ruixuan Li[†], Guoqiang Gao[†], Weijun Xiao[‡], Zhiyong Xu[⋆]

[†]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
[‡]Department of Electrical and Computer Engineering, University of Minnesota, Twin Cities, USA
[⋆]Department of Mathematics and Computer Science, Suffolk University, Boston, USA
Email: rxli@hust.edu.cn[†] ggq@smail.hust.edu.cn[†] wxiao@umn.edu[‡] zxu@mcs.suffolk.edu[⋆]

***Abstract*—In recent years, Peer-to-Peer (P2P) streaming systems experienced tremendous growth and became one of the largest bandwidth consumer on Internet. PPLive, one of the most popular applications in this category, is serving millions of registered users with hundreds of Live TV channels and millions of other video clips. Compared to the standard file sharing systems, the streaming service shows unique characteristics with more stringent time constraints and requires much higher network bandwidth. It is extremely important to evaluate and analyze existing applications, investigate the merits and weaknesses in these systems for the future development.**

In this paper, we conduct a comprehensive measurement study on PPLive. Both Live TV and Video-on-Demand (VoD) channels are evaluated. We record run-time network traffic on the client side, compare and analyze the characteristics of these channels based on their popularity. For both categories, we perceive that, in general, PPLive delivers satisfactory performance if enough concurrent peers are present in a particular channel. We also observe that VoD channels perform better than their counterparts in Live TV category in terms of data transmission, workload distribution, and signal traffic overhead. However, Live TV channels show better peer coordinations than VoD channels. Overall, our results reveal that although PPLive can provide excellent viewing experiences for popular channels, there are still challenges to fully support unpopular channels. New designs and algorithms are in urgent need, especially for unpopular Live TV channels.

I. Introduction

The last decade has witnessed the tremendous growth of Internet. According to the survey [1], from 1999 to 2009, Internet traffic increased from 32 PB per month in 1999 to 14,748 PB per month in 2009. The explosion was mainly due to popular distributed applications such as P2P file sharing [2], distributed computing [3], Internet phone service [4], Internet streaming [5], [6] and online social networks [7].

Among all these applications, Internet streaming and P2P file sharing systems are the killing applications which occupy the largest percentage of Internet traffic. Cisco [8] forecasted that global IP traffic will quadruple in the next four years and will exceed three-quarters of a zettabytes by 2014 and the video related applications (P2P and Internet streaming) will account for more than 91 percent of the total IP traffic.

Similar to P2P file sharing systems, Internet streaming applications also carry video traffic but have more stringent time constraints for data delivery. Viewers will experience severe watching disruption if data chunks can not be received before the deadline. Consequently, they demand higher Internet bandwidth. Based on the underlying infrastructure, Internet streaming applications can be classified into two groups: Client/Server (C/S) based and P2P based. YouTube [5] is the most famous application in the first category. It is a video-sharing website and the largest Internet streaming service provider. YouTube stores each video by a mini-cluster in its server farm. Popular videos are further replicated on multiple places in a content delivery network (CDN). All video watching traffic occur from YouTube or CDN servers to viewers. Clearly, the major challenge in this scenario is the scalability. The bandwidth provision, at video source servers or in CDNs, must grow proportionally with the client population. It results in very high bandwidth, hardware and power consumption expenses. Furthermore, due to the long tail effect, YouTube does not have excellent support for less popular videos.

P2P service model has been adopted to mitigate the load on central servers. In contrast to C/S based systems, P2P streaming applications require the minimum infrastructure support. Although not speak explicitly, P2P streaming applications are not purely server-less, a small group of dedicated data servers are furnished as the source servers. However, most data transfer operations are well disseminated among participating peers. Only a very small percentage of data transmissions occur from servers to viewers directly. The burden on servers is reduced significantly and there's no need to keep a large server farm or CDN as we do in C/S model. Therefore, P2P model is a promising alternative to C/S model in both cost and scalability. Figure 1 shows the differences of these two models.

The proliferation of large-scale P2P streaming applications have created the emergencies to understand the characteristic of those applications and attracted great interests from both industry and academia. However, we are facing with several challenges that stem from the commercial nature of these systems. The protocol/architecture/algorithm are proprietary, which makes it very hard to analyze the system and figure out the problems and bottlenecks. Most research studies used the black box approach to measure and evaluate system properties such as upload/download performance, user distribution, and session lengths, etc.

In this paper, we also take this approach and conduct extensive measurement study on P2P streaming applications. PPLive is chosen because of its popularity. As other studies, we capture network traffic traces on the client side using network monitoring software. The objective of our work is dissimilar from previous studies. We are interested in comparing the user behavior and performance characteristics of channels with different popularity. We are also interested in the fundamental feature differences of Live TV and VoD channels and their impacts on system performance. From the measurement data, we observe that for popular channels, there are abundant resources to support efficient streaming video delivery. While for unpopular channels, the lack of enough peer resources may result in sharp performance degradation.

In summary, we observe the following features for PPLive:

- For prevalent channels in both classes, PPLive can provide satisfactory performance. For unpopular Live TV channels, the viewers have possibilities to endure severe viewing problems.

978-0-7695-4393-2/11 $26.00 © 2011 IEEE
DOI 10.1109/CNSR.2011.12

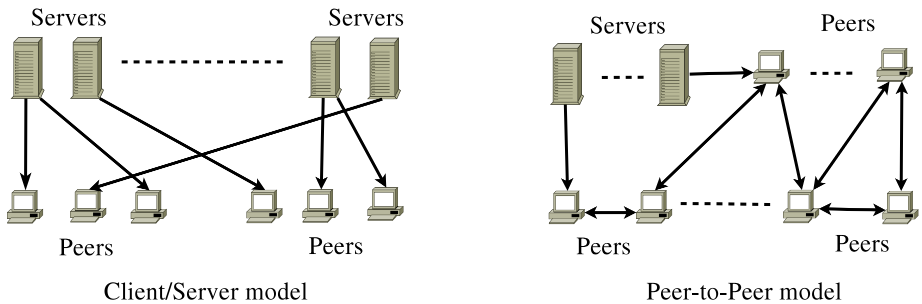

Fig. 1. Comparison of C/S and P2P models in streaming services

For unpopular VoD channels, such a problem rarely happens.

- The workload in most VoD channels is well balanced. For Live TV channels, the workload distribution is unbalanced. A small number of peers provide most video data.
- Overall, signal overhead is small in both categories. However, PPLive carries more signal traffic in Live TV channels than VoD channels.
- Live TV channels show better peer coordinations than VoD channels. In Live TV channels, more than 82.1% of upload traffic sent by our client goes to the peers from which it downloads data. While in VoD channels, this number is below 14.1%.

We believe those findings are very helpful for the development of future generations of P2P streaming applications.

The rest of the paper is organized as follows. In Section II, we briefly describe PPLive, the P2P streaming application we used in our study. In Section III, we introduce measurement strategy and methodology. We also discuss the network traffic monitoring tool used in our study. In Section IV, We discuss the experiments we conducted and analyze the results. In Section V, we summarize previous investigation studies on P2P streaming applications. Finally, in Section VI, we present our conclusion and the future work.

II. Overview of PPLive

Most commercial P2P streaming applications such as PPLive [6], PPStream [9], UUSee [10] and SopCast [11], etc. use mesh-based P2P infrastructure. These systems are mostly China-based applications. They achieved great successes and each of them has millions of registered users. The practices of these systems proved that P2P is an effective solution to delivery video contents to millions of simultaneous viewers across the world in real time. P2P streaming applications are still in their infant era. The video quality is not very high, mostly encoded in 350 - 600Kbps, which is much lower compared with the traditional cable and satellite broadcasting services. However, the viewers in P2P streaming applications have more freedom to choose their interested programs. There are plenty of available video resources and it is keep growing very fast.

PPLive [6] was first developed in Huazhong University of Science and Technology in Dec. 2004. It is one of the earliest and the most popular commercial P2P streaming applications in China. Since its appearance, PPLive was undergoing fascinating growth. For instance, the number of visitors of PPLive website reached 50 millions for the opening ceremony of Beijing 2008 Olympics while the dedicated Olympic channel attracted 221 million of views in only two weeks. PPLive is renamed to PPTV in 2010. However, in our experiments, we are still using the old version PPLive software. To avoid confusion, in the rest of the paper, we keep PPLive as the name of this application. PPLive uses pull strategy. The internal implementation details are proprietary due to its nature of commercial software. The PPLive web site provides limited information about its video content distribution mechanism. Nevertheless, various web sites and message boards provide additional information. Here, we briefly discuss its mechanism. The details can be found in [12] and [13].

In PPLive, all the viewers of a channel form an overlay network. Like aforementioned other streaming applications, the content of each channel is divided into multiple sub-streams and further divided into chunks. A small number of dedicated video servers are deployed for initialization of video broadcasting. Users who are interested in a video stream often have broadband connections which are high enough for them to act as relay points and forward the video clips to other users. Peers serve each other by exchanging chunks of data periodically. This allows the system to scale with the number of peers involved in the communication.

PPLive rates channels according to the number of simultaneous viewers. The hottest channel is assigned with a 5-star grade, and the number of stars decreases as the popularity of a channel decreases. The most unpopular channel has only 1 star or even no visible star at all. For a single channel, the popularity fluctuates from time to time. In general, for any channel, it has the smallest number of viewers after midnight and in the early morning, and has the highest number of viewers in the early evening. However, based on our experiments, we observe that the relative popularity or grade positions among different channels are always stable. For example, Hunan TV is a Live TV channel whose popularity varies from 4 to 5 stars. Qinghai TV is an unpopular Live TV channel whose popularity varies from 0 to 2 stars. At any moment, Hunan TV is more popular than Qinghai TV. We also choose several other channels such as Anhui TV and Shandong TV. The popularity of these channels is between Hunan TV and Qinghai TV. Accordingly, in our experiments, we select Hunan TV as the most popular channel, Qinghai TV as the most unpopular channel, and other channels as medium popular channels.

For VoD channels, we use some TV Series channels. Those channels have multiple episodes and last for tens of hours in total. Their popularity is pretty stable and does not fluctuate as fast as Live TV channels. In general, when a new TV Series channel is released, it becomes a hot channel and its popularity does not degrade for several months. In our experiments, we choose a Korean TV Series: Canlandeyichan as the most popular channel and a Chinese TV Series Changhuijiakankan as the most unpopular channel. Two other channels are chosen as medium popular channels. We also conduct measurement study on movies (another type of VoD which has shorter length). Due to the space limit, we did not include the results in this paper.

PPTV releases new versions frequently in 2010. However, in our measurement study, we keep it stable without upgrading operations. The PPLive version we used in all our experiments is v2.4.2.

III. Measurement Setup

In this section, we first describe the trace collection tool used in our study. Then we introduce hardware settings of our measurement experiments. Finally, we illustrate the information about channels used in the study, and demonstrate system configurations when collecting traces.

A. WireShark

WireShark [14] is a free and open source network package analyzer. It uses pcap API to capturing network traffic. WireShark is similar to tcpdump [15], except it has a graphic front-end interface. WireShark is easy to use and set up different configurations. We can create a filter to ignore uninterested data packets, decide the time duration to capture packets, and choose single file or multiple files to store the data. It is adequate for our project. In this study, we use WireShark for trace collection.

B. Hardware Settings

The experiments were carried out at Huazhong University of Science and Technology in Wuhan, Central China. We used multiple machines for trace collection. Each machine has Intel Xeon 2.5GHz CPU, 2GB memory, a 500GB hard disk, and a 100Mbps Ethernet card with windows XP Professional Service Pack 3 operating system installed. All the machines are connected directly to the largest ISP in China: China Telecom. The maximum outbound bandwidth is 2MB and shared by multiple computers. For each single machine, the in/out bound bandwidth is limited to 200KB only. Since most videos in PPLive are encoded around 400 - 650kbps, the in/out bandwidth for each machine is enough to support download/upload operations.

C. Experimental Configuration

In our experiments, we measure both Live TV and VoD scenarios. For each category, multiple channels are chosen. Table I shows the Live TV channels we use. For each channel, we collect traces for 15 hours time frame (we do not collect traces between 1 AM - 10 AM). As mentioned earlier, the popularity of a certain channel could change in this period, however, the relative hotness among channels does not change. For VoD channels, we choose episode-based TV Series channels. Each of them consists of multiple episodes with the number varies from 30 to 36. As Live TV channels, we choose TV Series channels with different popularity. Finishing all the episodes requires 24 hours to 30 hours. We truncate the data collection time to 24 hours. Table II shows the information about the TV Series channels we use. To simplify the demonstration, two letters are used to represent each channel. For example, Hunan TV is denoted as HN channel. Others are also denoted respectively.

TABLE I
LIST OF LIVE TV CHANNELS

Channel	Popularity(*)	Bitrate (kbps)	Trace Size	Date Collected
Hunan TV(HN)	3-5	396	1.55G	May 12 - 13
Anhui TV(AH)	2-5	390	1.42G	Jul. 10 - 11
Shandong TV(SD)	1-4	391	1.40G	Jul. 09 - 10
Qinghai TV(QH)	0-2	368	0.49G	Jun. 28 - 29

The steps when collecting traffic data of a certain channel are as follows: We open WireShark first, then we set up the configuration, such as the network protocol filter, location of the trace file, and the time span of data collection, etc. Next, we click the “start” button immediately although no PPLive traffic is available yet. The purpose to run it ahead is to avoid missing any PPLive traffic data. Finally, we open PPLive, and choose the channel we are interested in this run, and click watching. All network traffic will be recorded and stored in designated files. After the time period we set in WireShark expires, it stops trace collection.

TABLE II
LIST OF TV SERIES CHANNELS

Channel	Popularity(*)	Bitrate (kbps)	Trace Size	Date Collected
Canlandeyichan(CL)	4 - 5	511kpbs	2.26G	May 19 - 20
Xianhuaduoduo(XH)	2 - 4	632kpbs	1.78G	Jul. 24 - 25
Zhonganliuzu2(ZA)	1 - 4	447kpbs	1.27G	Jul. 25 - 26
Changhuijiakankan(CH)	0 - 2	535kpbs	1.15G	Jul. 22 - 23

Due to the large sizes of network packages and limited storage spaces, we do not keep entire packets. We only keep the first 150 bytes for each packet (including 42 bytes packet header and 108 bytes payload for UDP, 54 bytes packet header and 96 bytes payload for TCP). Since most signal packets are smaller than 150 bytes, we keep all the information inside. Only for video data packets which are larger than 150 bytes, we truncate the packet and throw the remaining data. Since all the important meta-data information are stored in the header, it does not affect the result analysis. By taking this approach, we can greatly reduce the storage overhead. We stored the truncated traffic data in contiguous trace files and set the size of each file as 100 megabytes.

All the measurement experiments were conducted between May 2010 and July 2010. We measured many channels in both types. The total size of the trace files is more than 50 gigabytes, and we only pick 4 channels with different popularity in each category. The trace files contain both TCP and UDP traffic, we ignore all the other types of network traffic since they are not related. The trace files are stored in pcap format. All the experiments were running under the default system configuration: network type is community broadband, maximum number of connections per channel is 30, and maximum simultaneous network connections is 20. When running an experiment, to avoid the disruption, we closed all the other network programs.

All measurement data processing was completed using Perl scripts and the figures were drawn using gnulpot [16] v4.2 patch level 6.

IV. EXPERIMENTAL RESULTS

We design several experiments to evaluate and compare Live TV and TV Series channels based on popularity. The metrics we inspected include the total amount of data received and sent, data transmission rate, number of connections, peer contribution breakdown, data signal traffic ratio, and peer receive/send data ratio, etc. In this section, we present detailed measurement information. To simplify the discussion, we use *client* to denote our machine used in the study.

A. Total Amount of Traffic

First, we examine the total amount of video data transmitted. Tables III and IV summarize the results for both Live TV and TV Series channels. In each table, the second and fourth columns show the sum of video traffic transferred for each channel. The third and fifth columns represent the percentage of data which dispatched using TCP.

In PPLive, video data can only be sent by TCP or UDP protocols. We discover that no matter it is a Live TV or TV Series channel, the vast majority of data is transmitted with UDP. This is mainly due to the high efficiency and small overhead of UDP. However, there’s one exception for the most unpopular TV Series channel CH. Almost three quarters of data it received uses TCP. We believe this is

TABLE III
TOTAL AMOUNT OF DATA TRANSFERRED FOR LIVE TV CHANNELS

Channel	Total Received(MB)	TCP (%)	Total sent (MB)	TCP (%)
HN	3121.1	0.05	1792.3	0.01
AH	2651.3	0.60	2136.3	2.05
SD	2711.2	0.90	1985.9	3.13
QH	789.9	0.68	458.9	0.02

because CH is so unpopular and very few peers can be explored. Our client has to retrieve most video data chunks directly from dedicated servers provided by PPLive. To ensure the quality, PPLive uses TCP protocol for data transmission. Furthermore, servers do not have to retrieve data from our client, thus the total amount of data sent in CH is the smallest among all channels.

TABLE IV
TOTAL AMOUNT OF DATA TRANSFERRED FOR TV SERIES CHANNELS

Channel	Total Received(MB)	TCP (%)	Total sent (MB)	TCP (%)
CL	5890.9	1.13	2276.5	0.01
XH	6618.5	0.76	633.1	0.02
ZA	4712.3	0.66	567.8	0.02
CH	6094.4	73.13	101.76	0.09

One thing that we have to point out here is, for TV Series channels, the total size of received data for each channel is approximately equal to the length of playback period multiple by its bitrate. For Live TV channels, only HN, AH and SD hold this property. For QH, the lack of available peers makes our client experienced serious viewing problem. In more than half of the period, PPLive keeps buffering and still can not obtain enough data chunks on time to ensure normal playback. Therefore, the total received data size in QH is much smaller than other Live TV channels although the total trace collection time span is the same for these channels.

B. Download Data Transmission Rate

The goal of download data transmission rate experiments is to obtain a snapshot of system throughput over time. In these experiments, we divide playback period into 1-minute slots. In the results, 42 bytes UDP headers and 54 bytes TCP headers are excluded. However, the application layer data header (used by PPLive itself) is included because we have no idea about its size. Figure 2 shows the results for Live TV channels. To make the figure more clear, we only keep the data between 18:00 and 22:00. The other time spans show similar results. From the figure, we observe that for HN, AH and SD, the download rates are stable. However, for the most unpopular channel QH, the download rate fluctuates from time to time, and in most time period, it is not able to obtain sufficient video data. Obviously, this is caused by the lack of qualified peers to support smooth data transmission. Overall, PPLive provides satisfactory performance for the most and medium popular Live TV channels since there are enough qualified peers in the overlay. For unpopular channels, the performance is poor.

Figure 3 shows the results for TV Series channels. As the previous experiments, to make it clear, we do not include all time slots. We choose the time span between 1000 and 1060 minutes from the start of the first episode. The curves in other time spans have the same feature and ignored. From the results, we learn that TV Series channels have completely contradictory behavior. The data transmission rates are very undulant. A high data transmission period is always companioned with a low or even no data transmission period. Apparently, this is because of the nature of TV Series. There's no synchronization requirement among the peers who are watching the same channel. The data of an episode can be cached on the peers who have watched recently. When another peer requests the data, it can be retrieved from an available peer without delay, and even the data which displaying time is not reached yet can be transmitted in advance. Thus, if network connection is stable, on the client side, a peer can download a large amount of data in a short time span. The peaks in the figure represent those time spans. When the buffer is full, our client downloading activity stops. There's few receiving data as shown in troughs in the figure. Until certain amount of video data has been consumed and the buffer is available again, the downloading process will resume. In contrast, for Live TV channels, it is no use to cache too much outdated video content. When viewing a Live TV channel, our client has to keep the pace with other peers and can only download a small amount of the most recent video content, the buffer can never be filled up.

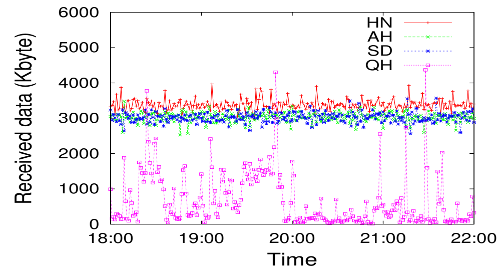

Fig. 2. Download Data Transmission Rates for Live TV Channels

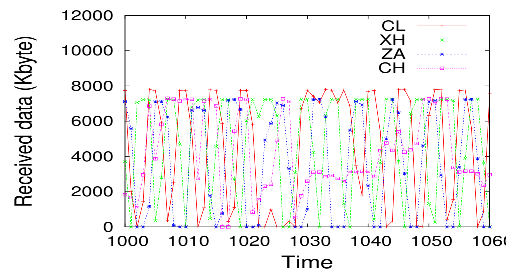

Fig. 3. Download Data Transmission Rates for TV Series Channels

For the most unpopular TV Series channel CH, the download data transmission rate behaves differently. When the network connection is stable and there are enough peers or dedicated servers to support data transmission, the curve is undulant as other channels, as shown between time span 1000 - 1028 in figure 3. If the network condition is not stable or there are no enough resources, our client has to download data contiguously from the dedicated servers with limited rates as we can observe between 1028 - 1048. However, unlike the most unpopular Live TV channels, with the support of dedicated servers, our client can still download enough video data to ensure smooth playback. Very few playing jitters appear.

C. Number of Peers Connected

During the playback, our client connects with many other peers for video data and signal message exchanges. The number of distinct peers our client has ever contacted can be used to represent the hotness of channels somehow. In this set of experiments, we count this number and the results are depicted in Tables V and VI. The second and fourth columns represent the number of distinct peers our client has video data or signal message connections with. The third and fifth columns only keep the peers with which our client has

TABLE V
TOTAL NUMBER OF CONNECTED PEERS FOR LIVE TV CHANNELS

Channel	Received Total	Received Data	Sent Total	Sent Data
HN	28313	3154	39985	4082
AH	7494	1797	11029	3083
SD	5647	1109	8530	2108
QH	11588	1982	14538	513

TABLE VI
TOTAL NUMBER OF CONNECTED PEERS FOR TV SERIES CHANNELS

Channel	Received Total	Received Data	Sent Total	Sent Data
CL	10520	4254	25073	2597
XH	3721	2335	6838	334
ZA	1601	1105	2250	204
CH	305	156	528	33

video data exchanges. For both Live TV and TV Series channels, the more popular the channel, the larger it has the number of peers to which our client receiving and sending data. The only exception is QH, the most unpopular Live TV channel. It shows more peers contacted than AH and SD. The explanation is that, it is hard to find the qualified neighbor peers for data exchanges. Consequently, PPLive scheduling algorithm keeps trying to contact other peers in the overlay. However, still no enough neighbor peers can be found to sustain smooth communications as we observed in previous sections. For all the other channels, such a problem does not exist.

Furthermore, we can examine that TV Series channels have smaller numbers of connected peers compared with their counterparts in Live TV category. The nature of TV Series channels makes caching an effective mechanism. On average, our client can always retrieve much larger data portion from a single peer than in Live TV channels. The total number of connected peers needed to support smooth playback in TV Series channels is much smaller than Live TV channels. PPLive scheduling algorithm does not perform neighbor peer searching actively.

D. Peer Connection Distribution

In this set of experiments, we measure peer connection distribution. We divide playback period into 1-minute slots as previous experiments. Then, we count the number of peers which have video data or signal exchanges with our client in each slot. The result for Live TV channels is shown in Figure 4. As we can observe, the number of peers connected fluctuates over time and it is relatively large. For HN channel, the number ranges from 600 to 1684. For AH and SD channels, the range is 80 to 613. Only for QH channel, the average number is smaller, but still the maximum number of peer connections could reach as high as 846.

Figure 5 shows the results for TV Series channels. The first impression is the number of peer connections is much smaller than Live TV channels. Even for the most popular channel CL, the number of peer connections varies between 8 to 449 with the majority in the range of 40 to 150. Obviously, such a phenomenon matches the explanations in data transmission rate experiments.

From the results, we can draw the conclusion that, due to the nature differences of Live TV and TV Series channels, they show different characteristics. TV Series channels keep a smaller number of peer connections than Live TV channels, but they retrieve more data from each peer they contacted.

E. Peer Connection Distribution for Video Data

In previous experiments, we count the peers which have either video data or signal exchanges with our client. To better understand

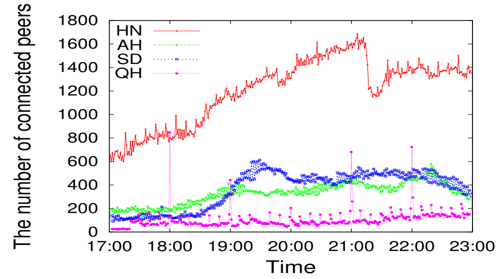

Fig. 4. Peer Connection Distribution for Live TV channels

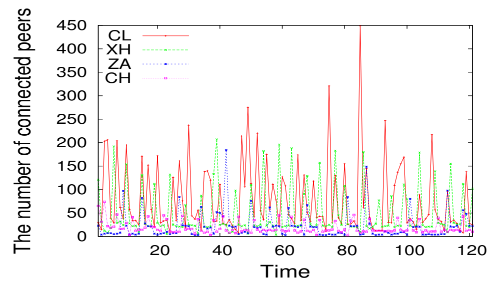

Fig. 5. Peer Connection Distribution for TV Series channels

the PPLive scheduling behaviors, we also calculate the number of peers which have the actual video data transmissions with our machine. We divide playback period into 5-minute slots. Signal message exchange only peers are excluded. The results are shown in Figures 6 and 7. Compared to the previous experiments, we observe that at any moment, both Live TV and TV Series channels only receive data from a very small set of neighbor peers (at most 180 peers, and most time in the range of 40 - 60 peers) regardless of the channel popularity. We calculate that video data can be supplied with such a number of peers is approximately equivalent to the number of chunks needed to support several minutes video playback. The minimal request unit for PPLive is one chunk, thus no more peers are needed for data transmission during this period. PPLive does not switch data exchange tunnels frequently to avoid high context switch overhead.

Due to the more stringent time synchronization requirement, Live TV channels have to keep tracking the information of more candidate peers as shown in our experiments. In case the communication tunnel to a current data exchanging peer becomes unstable, another peer can replace its position immediately to avoid performance degradation. However, most of them are used as backup only, no actual data transmission occurs. New designs might needed to reduce this overhead in Live TV channels.

F. Peer Contribution Breakdown

Another important metric we want to evaluate is the Cumulative Distribution Function (CDF) for peer contribution. We count all the peers from which our client downloads data from (we count the data package size larger than 1000 bytes as data packets, same as other research works do), as well as the peers to which our client uploads data. Figure 8 shows the results for Live TV channels. To make it clear, we only list the first 1000 peers. As we can see on the left, the most popular channel shows the best load balancing result. The first 20 most downloaded peers account for only 53 percent of total download traffic. For the most unpopular channel QH, it is 71 percent. Surprisingly, AH and SD have higher values (75 percent for AH and 77 percent for SD) than QH which means their downloading traffic is

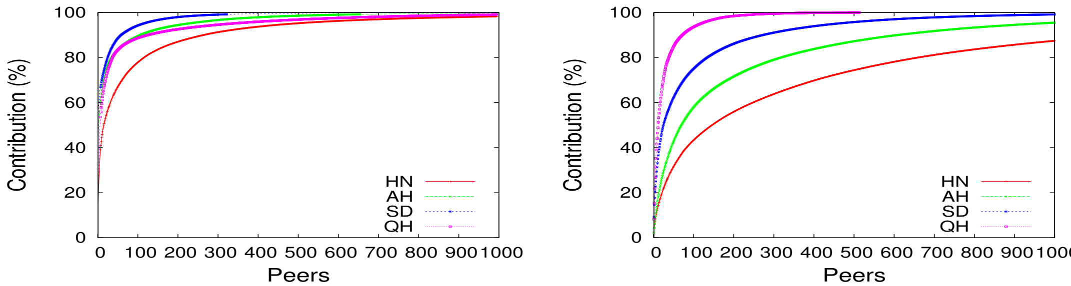

Fig. 8. Live TV channel Traffic CDF (Left: Receive, Right: Send)

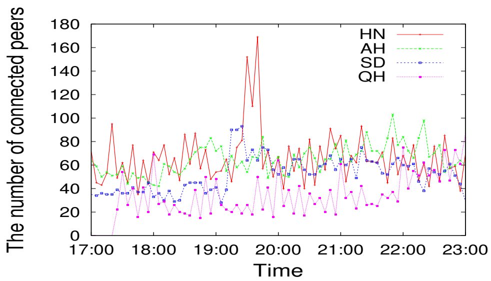

Fig. 6. Peers with Data Transmission Distribution for Live TV channels

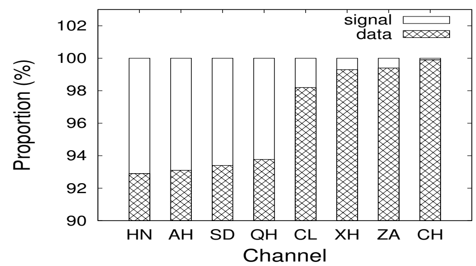

Fig. 10. Signal/Video Data Traffic Ratio

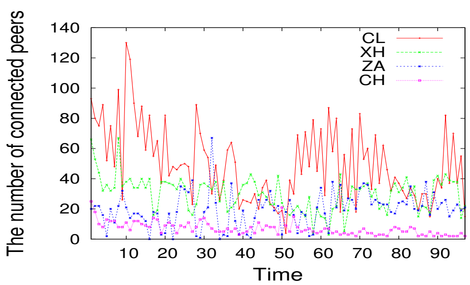

Fig. 7. Peers with Data Transmission Distribution for TV Series channels

even more concentrated. Clearly, PPLive scheduling algorithm always tries to choose the best peers for retrieving services.

On the right side on Figure 8, we can see the CDFs for upload traffic. For QH, the upload traffic is biased as in download situations. While for other channels, the upload traffic is better balanced. HN has the best performance with the top 20 most uploaded peers account for only 19 percent of total upload traffic. Obviously, PPLive aims to disseminate the workload evenly. For popular channels, a large number of peers can be chosen. For unpopular channels, the selection is limited.

Figure 9 shows the curves for TV Series channels. As we can see, for most channels, the load is well distributed on a large number of peers. They show better performance than their corresponding partners in Live TV channels. For example, for the most popular channel here, the first 20 most downloaded peers only account for 6.4 percent of total download traffic. It is much smaller than the most popular Live TV channel. For other channels, the same theory holds as well. The only exception is for the most unpopular channel, the total number of download peers is only 156, and the first 20 accounts for 85 percent. As we explained earlier, most receiving data communication is from dedicated servers with TCP. For upload services, TV Series channels show worse upload traffic balancing performance compared to their download distribution. In fact, the results are even worse than their counterparts in Live TV category. More research work has to be done to better understand this phenomenon.

G. Singal and Video Data Traffic Ratio

We also check the ratio of signal and video data traffic for all the channels. The results are presented in Figure 10. We observe that both categories show pretty good performance with signal traffic is much less than video data traffic. However, TV Series channels are more efficient than Live TV channels. For Live TV channels, the ratio is stable regardless of popularity. On average, about 93.3% traffic is actual data traffic. For TV Series channels, the value is 99.2%. This is because in Live TV channels, peers connect to more peers than TV Series channels and they have to exchange the buffer bitmaps frequently. Thus, more signal traffic is generated. For TV Series channels, most data are served from the cached partition on neighbor peers, thus a single bitmap signal message contains information of a large number of data chunks. It is not necessary to exchange this information as frequently as Live TV channels. Surprisingly, the most unpopular channel CH shows the best efficiency. This is understandable as we described in the previous sections, most video data transmission for CH is from the dedicated servers with TCP connections, and very few signal messages are needed.

H. Peer Receive/Send Data Ratio

The principle of P2P paradigm is to build a tight cooperation relationship among participating peers. We analyze PPLive performance in terms of peer receive/send data ratio. First, we list the top 100, 500 and all peers from which our client has received video data. Then, we count the ratio of the total data our client sends to these peers over all sent traffic. The results are shown in Figure 11. Live TV channels exhibit excellent peer coordination performance. All four channels have higher ratios than TV Series channels. Among them, the most unpopular channel QH has the best effect. About

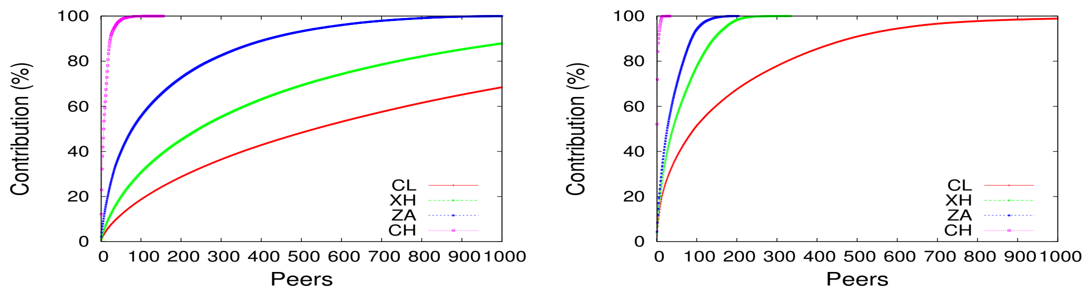

Fig. 9. TV Series channels Traffic CDF (Left: Receive, Right: Send)

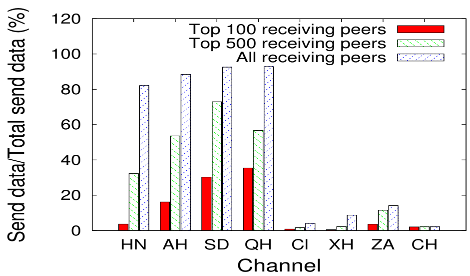

Fig. 11. Peer Receive/Send Data Ratio

35% of sent data goes to the top 100 peers our client has received data from. More than 90% of sent data goes to the peers our client ever received data from. There are two factors for that. One is small number of total peers for QH, and the other is PPLive's preference of selecting qualified neighbor peers. For unpopular Live TV channels, it is not easy to find good neighbor peers for cooperation. When such a peer is found, the system keeps exchanging data as long as possible. While for popular channels, sufficient candidate peers exist. To achieve better load balancing, the system does not want to keep a connection too long. Overall, Live TV channels perform very well with 82.1% to 92.8% of total sent data going to the receiving peers.

We can also observe that TV Series channels show opposite performance. There's very few peer cooperation. For all channels, only 2.2% to 14.1% of sent data traffic goes to the peers from which our client have received data. In these channels, when our client watches an episode, it always retrieves the video content from another peer who has already watched this part and cached it on the disk. Most likely, this peer is way in advance. The video content it needs can not be retrieved from our client. It can only request from other peers who are watching ahead of him. Thus, very few data exchange occurs. This problem is even worse for the most unpopular channel CH since its data is mainly retrieved from the dedicated servers. Servers do not retrieve data from peers. This scenario does not mean bad peer resource utilization. It only represent that, in TV Series channels, the group of peers our client receiving data from has little overlap with the group of peers our client sending data to. Some mechanisms can be used to improve the performance such as prefetching. We plan to do further investigations on this issue.

I. *Other Metrics*

We also evaluate other metrics such as packet size distribution, peer location distribution, session numbers and session lengths. Due to the limited spaces, we do not include the results here.

V. Related Works

To better understand user behaviors, network characteristics and performance bottlenecks in P2P streaming applications, numerous research papers have been published to evaluate and analyze these bandwidth intensive applications [17], [18], [19].

In [20], Y. Liu, et al. conducted a survey on the existing P2P streaming technologies for both Live TV and VoD services. Key designs, including system topology, peer connection and management, as well as data transmission scheduling mechanism were discussed. They described solutions as well as challenges of providing streaming services in a P2P environment. All the tree, multi-tree and mesh-based infrastructures were introduced. They concluded that although video qualities are still not comparable with the traditional TV service providers, P2P streaming applications already become a serious concern for ISPs, and efficient solutions are desired to reduce the heavy traffic burden.

[21] conducted an experimental comparison of two streaming applications (a mesh-based and a multiple tree-based). This was the first study to compare streaming overlay architectures in real Internet settings, considering not only intuitive aspects such as scalability and performance under churn, but also less studied factors such as bandwidth and latency heterogeneity of overlay participants. In their study, the authors found that mesh-based approach is superior than tree-based approach. Their conclusion could explain why most commercial P2P streaming applications are using mesh-based infrastructure. However, they conducted the controlled small-scale measurement on PlanetLab [22], which only has a small number of concurrent users, and they chose Chainsaw [23] and SplitStream [24] systems which were not the real world commercial systems.

Hei et al. [12] conducted one of the earliest measurement studies on commercial P2P streaming applications. The experiments were also based on PPLive. Ethereal [25] was used to collect traces of Internet traffic on two channels CCTV-3 and CCTV-10 for about two hours. They observed that PPLive provided very satisfactory performance. Their work was on Live TV channels only and no VoD channels have been chosen. They did not consider multiple channels of different popularity and the lengths of trace data collection was short which might not be enough to fully represent the system behavior.

Vu et al. undertook a crawler based investigation on PPLive [13]. The results have shown that many well-known conclusions for P2P overlays become false when one considers media streaming services atop the overlay instead. They observed that PPLive users are impatient, channel size variations are larger, average degree of a peer in the overlay is independent of channel sizes. Their work was concentrated on the characteristic differences between normal P2P file sharing systems and P2P streaming applications. Only three channels were used and they were popular channels with many viewers, no unpopular channels were considered. Furthermore, all three were VoD channels, no Live TV channels were chosen.

In [26], Liang et al. conducted a measurement study on another popular P2P streaming system, PPStream. Their study was carried during the Beijing 2008 Olympics broadcasting, which was a large-scale event with a huge amount of live interests. Hence, their work focused on the PPStream capability and scalability of dealing with a large number of simultaneous users on a very hot channel. It has already proved that P2P streaming applications can achieve satisfactory performance in such a scenario.

Ciullo, et al. [27] conducted a controlled measurement evaluation on PPLive and Joost [28]. The authors evaluated the characteristics of both data distribution and signaling process for the overlay network discovery and maintenance. By considering two or more clients in the same sub-network, they observed the capability of both systems to exploit the locality of peers. They also explored how the system adapts to different network conditions.

Kermarrec et al. [29] investigated the channel switching latency and overhead issues in PPLive. Their measurement revealed the most important operations in this process, and they proposed a simple distributed algorithm to leverage the overhead and speedup this switching process. The solution seems to be appealing for P2P streaming applications. However, the lack of consideration on channel popularity could reduce its effectiveness and may bring in unnecessary maintenance traffic.

Our work distinguishes from all the existing measurement studies in the following aspects. We focus on the measurement and analysis of the user behaviors, network traffic and performance differences among channels with various popularity. We run our experiments for a long period (at least 15 hours) to reflect the dynamic nature of P2P streaming systems and achieve an unbiased evaluation results. Furthermore, we conduct the study on both Live TV and VoD channels, and discover fundamental differences between these two scenarios.

VI. Conclusions and Future Work

In this paper, we conducted a measurement study on a popular online streaming application PPLive. We have gained valuable insight to the behaviors of these applications by collecting and analyzing the data and signal traffic in various scenarios. It helps us identify the determinative factors affecting the performance in P2P streaming applications. We measured and evaluated system characteristics such as data transmission rate, peer contribution distribution, signal/data traffic ratio, peer coordination performance of Live TV and VoD channels with different popularity. According to our experimental results, we concluded that PPLive provides different levels of QoS support for various channels. In general, popular channels often have sufficient resources for smooth playback. While the lack of resources makes unpopular channels experiencing serious issues, especially for Live TV channels where caching mechanism is not effective. The viewers of unpopular channels are suffering from occasional glitches, re-buffering and broken streams, annoying hiccups, substantial delays and latencies. Clearly, a better scheduling and resource distribution algorithm is needed to solve this issue. We believe our findings give the research and industry communities better understanding about the P2P streaming applications and provide some new insights for designing and optimizing new generations of P2P online streaming applications.

In the future, we will continue our measurement study on online streaming applications. We will choose other applications such as PPStream, UUSee and SopCast and compare them with PPLive. We plan to develop new caching and relay distribution algorithms to improve the performance of unpopular channels as well, especially for Live TV channels.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant 60873225, 60773191, 70771043, National High Technology Research and Development Program of China under Grant 2007AA01Z403, Natural Science Foundation of Hubei Province under Grant 2009CDB298, Wuhan Youth Science and Technology Chenguang Program under Grant 200950431171, Open Foundation of State Key Laboratory of Software Engineering under Grant SKLSE20080718, Innovation Fund of Huazhong University of Science and Technology under Grant Q2009021.

References

[1] Miniwatts Marketing Group, "Internet World Stats, http://www.internetworldstats.com/stats.htm."
[2] BitTorrent, "http://www.bittorrent.com."
[3] SETI@HOME, "http://setiathome.ssl.berkeley.edu."
[4] SkyPe, "http://www.skype.com."
[5] Youtube, "http://www.youtube.com."
[6] PPLive/PPTV, "http://www.pptv.com."
[7] Facebook, "http://www.facebook.com."
[8] Cisco Inc., "Cisco visual networking index: Forecast and methodology, 2009-2014."
[9] PPStream, "http://www.ppstream.com."
[10] UUSee, "http://www.uusee.com."
[11] SopCast, "http://www.sopcast.com."
[12] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "Insights into pplive: A measurement study of a large-scale p2p iptv system," in *Proceedings of IPTV Workshop, International World Wide Web Conference*, (Edinburgh, Scotland), May 2006.
[13] L. Vu, I. Gupta, J. Liang, and K. Nahrstedt, "Mapping the pplive network: Studying the impacts of media streaming on p2p overlays," tech. rep., Computer Science Department, University of Illinois at Urbana-Champaign, August 2006.
[14] Wireshark, "http://www.wireshark.org."
[15] Tcpdump, "http://www.tcpdump.org."
[16] GnuPlot, "http://www.gnuplot.info."
[17] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A measurement study of a large-scale p2p iptv system," *Multimedia, IEEE Transactions on*, vol. 9, pp. 1672–1687, November 2007.
[18] S. Ali, A. Mathur, and H. Zhang, "Measurement of commercial peer-to-peer live video streaming," in *Proceedings of ICST Workshop on Recent Advances in Peer-to-Peer Streaming*, (Weaterloo, Canadda), August 2006.
[19] S. Zeadally, E. Cerqueira, M. Curado, and M. Leszczuk, "Session level analysis of p2p television traces," in *Proceedings of Third International Workshop on Future Multimedia Networking (FMN)*, (Krakow, Poland), June 2010.
[20] Y. Liu, Y. Guo, and C. Liang, "A survey on peer-to-peer video streaming systems," *Peer-to-Peer Networking and Applications*, vol. 1, pp. 18–28, March 2008.
[21] J. Seibert, D. Zage, S. Fahmy, and C. Nita-rotaru, "Experimental comparison of peer-to-peer streaming overlays: An application perspective," tech. rep., Purdue University.
[22] "PlanetLab: An open platform for developing, deploying and accessing planetary-scale services." "http://www.planet-lab.org".
[23] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, A. E. Mohr, and E. E. Mohr, "Chainsaw: Eliminating trees from overlay multicast," in *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS)*, (Ithaca, NY), pp. 127–140, Feburary 2005.
[24] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "Splitstream: high-bandwidth multicast in cooperative environments," in *Proceedings of the nineteenth ACM symposium on Operating systems principles (SOSP)*, (Bolton Landing, NY), pp. 298–313, October 2003.
[25] Ethereal, "http://www.ethereal.com."
[26] W. Liang, J. Bi, R. Wu, Z. Li, and C. Li, "On characterizing ppstream: Measurement and analysis of p2p iptv under large-scale broadcasting.," in *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, (Honolulu, Hawaii), pp. 1–6, November-December 2009.
[27] D. Ciullo, M. Mellia, M. Meo, and E. Leonardi, "Understanding p2p-tv systems through real measurements," in *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, (New Orleans, LA), pp. 2297–2302, November-December 2008.
[28] Joost, "http://www.joost.com."
[29] A. M. Kermarrec, E. Merrer, Y. Liu, and G. Simon, "Surfing peer-to-peer iptv: Distributed channel switching," in *Proceedings of the 15th International Euro-Par Conference on Parallel Processing (Euro-Par)*, (Delft, The Netherlands), pp. 574–586, August 2009.