Performance of P2P Live Video Streaming Systems on a Controlled Test-bed

Sachin Agarwal   Jatinder Pal Singh

Deutsche Telekom A.G.,
Laboratories
Ernst-Reuter-Platz 7
10587 Berlin, Germany
{sachin.agarwal,jatinder.singh}@telekom.de

Aditya Mavlankar   Pierpaolo Bacchichet
Bernd Girod

Department of Electrical Engineering,
Stanford University
350 Serra Mall
Stanford, CA 94305, USA
{maditya, bacci, bgirod}@stanford.edu

ABSTRACT

We evaluate and compare the performance of three P2P streaming systems that are capable of streaming live video on the Internet by testing them on a carefully controlled, traffic-shaped network test-bed. We first describe the construction of the test-bed based upon Internet measurements between geographically distributed hosts. Then, we present a methodology for evaluating these P2P video streaming systems by performing video quality and network usage analysis from the log information obtained via running these systems on the test-bed. Our methodology to assess P2P live video streaming systems comprises analyzing the objective quality of the received video, waiting time to receive the first data byte, and several network usage measures such as P2P protocol overhead, load on the server due to the inefficiencies of the P2P overlay, and measurements of the number of bytes exchanged between the peers. It is essential that every peer buffers packets for some time before playing out the video in order to ensure good quality. We report the time that the user has to wait before he can see the video playing. These measurements not only gauge the performance of currently available P2P streaming systems but also highlight desired improvements in current P2P video streaming systems.

Keywords

Peer-to-peer, P2P, live video streaming, video quality, traffic-shaping, test-bed

1. INTRODUCTION AND MOTIVATION

Peer-to-peer (P2P) live and on-demand video streaming have become notable Internet applications offering diverse video content to millions of viewers, with many academic and commercial P2P systems having large installed user bases. Owing to the vast number of available systems, a common methodology for comparing the technical aspects of these systems is required to assess the quality of their underlying protocols and implementations. This work provides a well-defined test-bed setup and analysis procedure to perform a quantitative comparison of P2P video streaming systems based on several relevant parameters.

P2P video streaming systems are deployed over the best-effort Internet. Owing to the stringent QoS requirements of video streaming, P2P video streaming systems need to be tested in real-Internet like conditions for an accurate understanding of their capabilities. Therefore, we set up a carefully controlled emulation of the Internet on our test-bed using traffic shaping mechanisms to mimic Internet characteristics. In addition, we simulate peer behavior of joining and leaving the P2P session that mimics user behavior in real-world P2P systems. By setting up this controlled test-bed, we can ensure fairness in testing different systems. The traffic shaping and simulated peer joins and leaves can be easily modified in order to test the systems running over different types of networks, and with different user behavior patterns.

We have developed an experimental methodology to assess P2P streaming systems by deriving performance results from the logs obtained by running the P2P video streaming systems in the test-bed. Received video quality, the user's waiting time, video resolution, etc. are important factors which affect the popularity and adoption of P2P streaming systems. One of the key contributions of our work is in setting up an experimental framework for video quality measurement for P2P streaming systems.

Our experimental methodology provides several quality measures, like the Peak Signal to Noise Ratio (PSNR) [25] of the decoded video and the channel start-up time which is defined as the waiting time to receive the first byte of data. We also provide network efficiency characteristics like the ratio of P2P to server bandwidth used by the system, P2P protocol overhead, inefficiency due to duplication in the downloaded stream, aggregate bandwidth usage of the tested solutions and the packet loss experienced by the peer clients.

The paper is organized as follows. In Section 2 we review related work in P2P streaming measurements and video quality assessment. In Sections 3 and 4, we describe the design and setup of our experiments and the test-bed, including special video files that we generated for the analysis. The assessment of the P2P video streaming systems deployed on our test-bed is presented in Section 5. Section 6 summarizes our findings as well as provides future work directions.

2. RELATED WORK

The traffic characteristics of large-scale P2P streaming systems has been a topic of interest since the first P2P streaming systems [5–

TRIDENTCOM 2008, 17th – 20th Mar 2008, Innsbruck, Austria.

DOI 10.4108/icst.tridentcom.2008.3110

7, 19]. A survey and comparison of the approaches and algorithms employed in various P2P streaming overlays can be found in [16, 17]. There have also been recent studies of commercial P2P streaming systems (for example, [3, 11, 12, 26, 28–30]) that study networking characteristics of some commercial P2P systems such as SopCast [9, 27], PPLive [21], Coolstreaming [8, 31] and Gridmedia [10]. In all these studies, the focus is not on received video quality measures such as PSNR and video startup times and they are limited to an analysis of the system performance as a whole. For example, reference [24] is a recent measurement study on SopCast that reports an extensive list of metrics but does not include video PSNR. Instead the authors only consider pre-roll delay and a continuity index (which is determined by lost packets) that may not accurately indicate the loss in video quality because different packets may be less or more important depending on the type of video-frame data in the packet.

Video quality estimation of Internet streaming using video traces has been extensively studied and documented in [15, 22, 23] but without specific application to P2P streaming measurements. These techniques are particularly powerful since they translate network statistics, namely packet loss and arrival time, into video statistics, like PSNR and frame losses. A similar approach has been used in [4] to analyze the video quality provided by the Stanford P2P Multicast (SPPM) solution for experiments performed over the Planet-Lab test-bed [20] for 100 peers, but this setup was limited to high speed university connections with no peer churn.

To the best of our knowledge, this is the first head-to-head comparison of commercial P2P video streaming systems under homogeneous testing conditions. Most other measurement studies of P2P systems were done on the basis of logs that were collected from their deployment on the Internet. However, this entailed little possibility of a fair comparison between different systems owing to differences in infrastructure, network conditions, video characteristics, and client behavior. We have excluded the names of the tested P2P systems from this work in order to protect the commercial interests of the system providers.

3. TEST-BED

We next describe the setup of our test-bed comprising several clients and two servers that formed our traffic-shaped IP network used for testing the video streaming systems. The peer connections in the deployed network represent broadband Internet connections having heterogeneous bandwidths, delays, and packet loss rates (PLR). We further included some real DSL connections in our test-bed to incorporate real-world peer Internet connections. The network characteristics of the other representative peers were controlled through careful traffic shaping. We wish to emphasize that the choice of various network parameters in our test-bed could be suitably modified to emulate different network conditions.

For our experiments, we focused our efforts on the sub-hundred peer scenario where a user wishes to transmit special interest content at approximately SIF resolution (352×240 pixels) to tens of geographically distributed peers through an IP network such as the Internet. While the scale of our test-bed is not representative of very large scale P2P video streaming systems, the methods and analysis we report are equally applicable for such scenarios.

3.1 P2P Streaming Systems

The P2P streaming systems that we employed provide users with P2P client software and content providers with server software to receive and broadcast video streams. We evaluated these systems for several user centric quality measures. Since we had direct access to the server components of these systems and tested the systems under controlled and repeatable network conditions, we report fine-grained network behavior and video quality of the tested systems.

We obtained slightly modified versions of the clients with additional integrated logging features from the system providers in order to facilitate our analysis. The server and client software was operable from the command line in order to help scripting for test automation.

In order to protect the commercial interests of the 3 tested P2P video streaming systems, we present anonymized results by referring to the tested systems as System A, System B, and System C. System A was an overlay multicast tree-based P2P streaming system while systems B and C were mesh-based P2P streaming systems. All three systems are in an advanced stage of development and have been deployed successfully on the Internet. The modified versions were obtained in early 2007 and all results in this paper are indicative of the characteristics of these versions.

Table 1: NISTNet Network model: Average Delay (ms), Jitter (ms) and PLR between hosts at different locations measured using Abing, and used to configure the hosts in the test-bed via NISTNet.

	Delay	Jitter	PLR
server to Berlin	24.17	4.8	0.001
server to Stanford, Berlin to Stanford, & Munich to Stanford	109	24	0.001
server to Munich, & Berlin to Munich	29	4	0.0001
DSL to DSL	29	4	0.0005
within Munich	0.4	0.1	0.0001

3.2 Controlled Network Setup

We set up a controlled IP network to test the video streaming system under real-world network conditions (Fig. 1). Our test-bed consisted of 48 dedicated client PCs hosted in data centers in Berlin and in Erfurt, Germany. These two data centers were connected via a 52Mbps backbone link. Two servers in the Berlin data center hosted the P2P streaming servers. The 8 clients in Berlin were connected to the Internet via DSL connections provided by a major ISP. The client connections' upload speeds are indicated in Fig. 1.

The bandwidth, delay, jitter, and PLR of the client Internet connections were configured through traffic shaping using the NISTNet [1] tool to represent the real-world network characteristics measured between hosts in Berlin, Stanford, and Munich. For the emulation of network characteristics between different geographic locations, we first ran the Abing [18] tool repeatedly between computers connected to the Internet and located in the cities mentioned above. This yielded multiple RTT measurements that we used to compute delay (RTT/2) and jitter statistics using a Gaussian profile. These statistics were used to drive packet delay and delay jitter in NISTNet. The PLRs between different geographical locations were configured as per real-time measurements reported in [2]. Table 1 shows the summary of the network statistics as configured between the servers/clients at different locations. The upload bandwidth was configured via NISTNet to values indicated in Fig. 1.

All the client PCs were controlled from a central location in Berlin using the Hobbit Monitor tool [13]. Each P2P client was hosted on a virtualized OS instance (Linux or Windows, depending on the P2P client software's platform) running on a client PC whereas the NISTNet traffic shaper was installed directly on the client PC. The virtualized OS instance used its hosting client PC

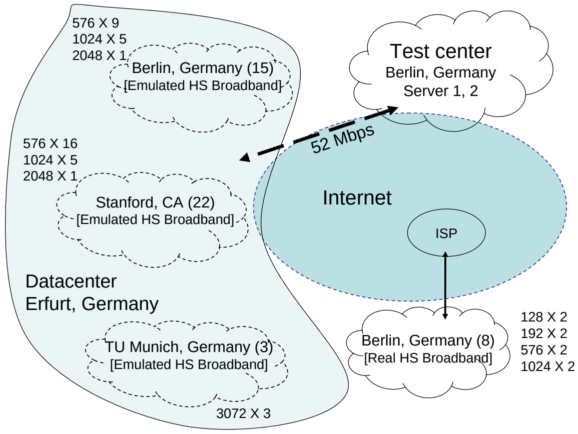


Figure 1: Network Setup: The physical and "emulated" network setup on the controlled network test-bed. The upload bandwidths b (kbps) and the number of clients n are marked as $b \times n$. The clouds represent emulated clients while the colored enclosures are the physical location of the hosts in the data centers.

as a network router, experiencing the desired traffic shaping induced by NISTNet. This virtualization also allowed us to deploy the tested systems on exactly the same physical infrastructure, thus eliminating the possibility of hardware characteristics introducing any bias in the results.

Several other features such as automated collection and analysis of logs and periodic re-confirmation of bandwidth, delay, and delay jitter were built into the system to ensure smooth operation.

3.3 Peer Characteristics

Each P2P client was operated according to an On-Off model that emulates peer churn (peers joining and leaving) for the P2P streaming system in our test-bed. During each 6 minutes time slot durations during the test-runs, a client is on or off with probabilities 0.9 and 0.1 respectively. Also, a client could switch off for the rest of the run at any time slot with probability 0.05. In order to emulate the end-game characteristics when peers rapidly depart the P2P overlay at the end of the live stream's transmission, each client left with a probability of 0.5 in the last 5 minutes of the run.

4. QUALITY EVALUATION

4.1 Video Bit-Stream Employed

We encoded 30 minutes of the classic movie *La Dolce Vita* (Fellini, 1960), using a state-of-the-art H.264/AVC [14] video codec with accurate rate-control to generate a bit-stream with a constant bitrate of 400 kbit/s. The spatial resolution of our test video is 352×240 pixels and the frame-rate is 24 fps and the average PSNR of the encoded sequence is approximately 42 dB. We omit the audio stream and only report video quality measurements in this work.

In order to allow users to tune in any time during the streaming session, an intra-coded picture (I) is inserted every second, thus the group of pictures (GOP) is 24 frames long. The number of consecutive bi-directionally predicted (B) frames is two. The unidirectionally predicted (P) frames use a single previous frame for reference, hence decoding can be synchronized starting from an I frame. A start-code of three bytes allows to detect the boundaries of every encoded frame. We use the ASF (Advanced Systems Format) container for wrapping the H.264/AVC coded stream for P2P Systems B and C since these systems make use of the ASF format for parsing the bit-stream and extracting useful timing information. System A parses the H.264/AVC bit-stream itself and extracts the timing information.

4.2 Video PSNR

A best-effort packet-switched network entails delay, delay jitter and packet loss (which may be due to network packet loss or P2P streaming issues like peer churn). This means that some parts of the bit-stream never arrive or arrive too late for playout and this affects the quality of the displayed video. For the lost portions of the bit-stream, the video decoder employs error concealment to limit the quality degradation of the displayed video. For our experiments, we assume that the video decoder uses "Copy Previous" error concealment, i.e., it replaces lost portions of an image with the corresponding regions from the previously decoded frame. In order to simplify the video quality assessment, we assume that a packet loss associated with a frame causes the loss of the whole frame. A video frame is not decodable, and hence considered to be lost, if either this frame or any other frame that this frame depends on are lost. If a previously decoded frame is displayed in lieu of the current frame then the display seems to have frozen and this is called a frame freeze. The loss of a large portion of contiguous data causes the loss of several consecutive frames and leads to a long frame freeze.

When the peer's video display is on, for every frame-interval, we estimate the quality of the displayed video frame by computing the PSNR between the original uncompressed video frame and the frame which is actually displayed according to the concealment algorithm described above. If a frame is completely decodable then the PSNR only depends on the distortion due to quantization induced at the encoder, whereas a frame-freeze causes the PSNR to drop steadily as the dissimilarity between the original uncompressed frame and the displayed frozen frame increases.

In order to translate the information about the loss of packets into the loss of frames, we utilize the knowledge about the location of the encoded frames within the streamed file, the frame dependencies as well as the frame display deadlines. The frame display deadlines are decided by the buffering time and the frame-rate of the video.

4.3 Startup Delay

In the live streaming scenario under investigation, any peer that joins when the session is underway, should ideally start receiving the video from the current video frame. However, it should be noted that the peer initially spends some time in getting connected to the content distribution pool. This introduces some startup delay, also called as initial connection time for the peer. We measure the startup delay as the time required by the peer to receive the first byte of data after initiating the connection to the session.

Although the startup delay indicates how fast the P2P protocol can get a peer connected to the data distribution pool, every peer needs to buffer packets for some time before playing out the video in order to sustain good quality through the session. The waiting time from connection initiation until the playout is called pre-roll delay or buffering time. We report this quantity too. It should be noted that although a given system might have low startup delay, it could still require long buffering time. It was observed that this is true of the mesh-based protocols since the advertising and delivery of some chunks of data can consume a lot of time compared to other chunks.

4.4 Server to P2P Bandwidth Usage

The commercial value of P2P streaming for a content provider is that the bandwidth costs can be reduced by using the uplink bandwidth of peers to distribute content. Therefore, an interesting measure is the comparison of the amount of the video stream downloaded by a P2P client from the server to that downloaded from other P2P clients. Despite the well-provisioned server uplink bandwidth in our test-bed, some tested P2P video streaming systems mostly use the P2P bandwidth in order to keep the server bandwidth usage low.

4.5 Total Received Bytes and Duplication among Received Packets

We report the total number of bytes received at the PC's network interface at each client as a percentage of the size of the video stream required to perfectly play back the video during the on-time of the peer. Note that this includes the duplication among received packets. Hence, we also report the amount by which the received stream falls short as a percentage of the size of the video stream required to perfectly play back the video during the on-time of the peer. There is good correspondence between the volume of the missing portion of the bit-stream and the drop in the video PSNR.

4.6 Transmit/Receive Footprint

Finally, we show the break-up of the total bytes received at each peer from other peers during the P2P streaming session. This yields valuable insights into the robustness and efficiency of the P2P protocol. For example, System A, being a tree-based protocol, tends to send more information along fixed routes as compared to the data driven protocols implemented by Systems B and C. This characteristic of the latter protocols is responsible for greater robustness, at the cost of more duplicate blocks.

5. EXPERIMENTS

We completed two types of test runs with the P2P streaming systems for streaming the 30 minute long video file to 48 clients (client IDs $1 \ldots 48$). Run 1 corresponds to cases where the network was traffic-shaped using NISTNet according to the model presented in Section 3 whereas Run 2 corresponds to cases where the NISTNet traffic shaping was disabled and the physical network characteristics were applicable. The On-Off model described in Section 3 was active in both runs.

Each run was repeated multiple times for the three systems tested and the results presented here are compiled from the best performances in each type of run. The best runs are decided according to the average PSNR across all clients. In the presented results for any run, we avoided taking the statistical average of the trials because averaging quality across multiple trials would hide some of the adverse effects that users may otherwise observe in video playback. These best trials are nevertheless representative of the other trials owing to the low variance in performance across the different trials that we carried out on our controlled test-bed.

5.1 Experimental Analysis Results

We start by analyzing the bandwidth efficiency of the tested solutions as described in Section 4.5. We first define video stream as the bytes fed to the media decoder by the P2P client running on a peer. These data are stripped off of any protocol control packets, duplicates, headers, etc. and only comprise of video data useful to the media decoder; in our case this is the H.264/AVC stream in its appropriate media container such as Microsoft's ASF (Advanced Systems Format).

Figure 2 shows the bandwidth efficiency of Systems A, B and C for Run 1 and Run 2 respectively. In particular, the white bars show the number of bytes received at the network interface of each client as a percentage of the size of the video stream required to perfectly play back the video during the on-time of the peer. The filled bars in the figures indicate the percentage of bytes actually served to the media decoder; 100% would indicate that all the required video stream was served to the media decoder. The more the filled bars fall short of the dotted line the worse is the received video quality in general.

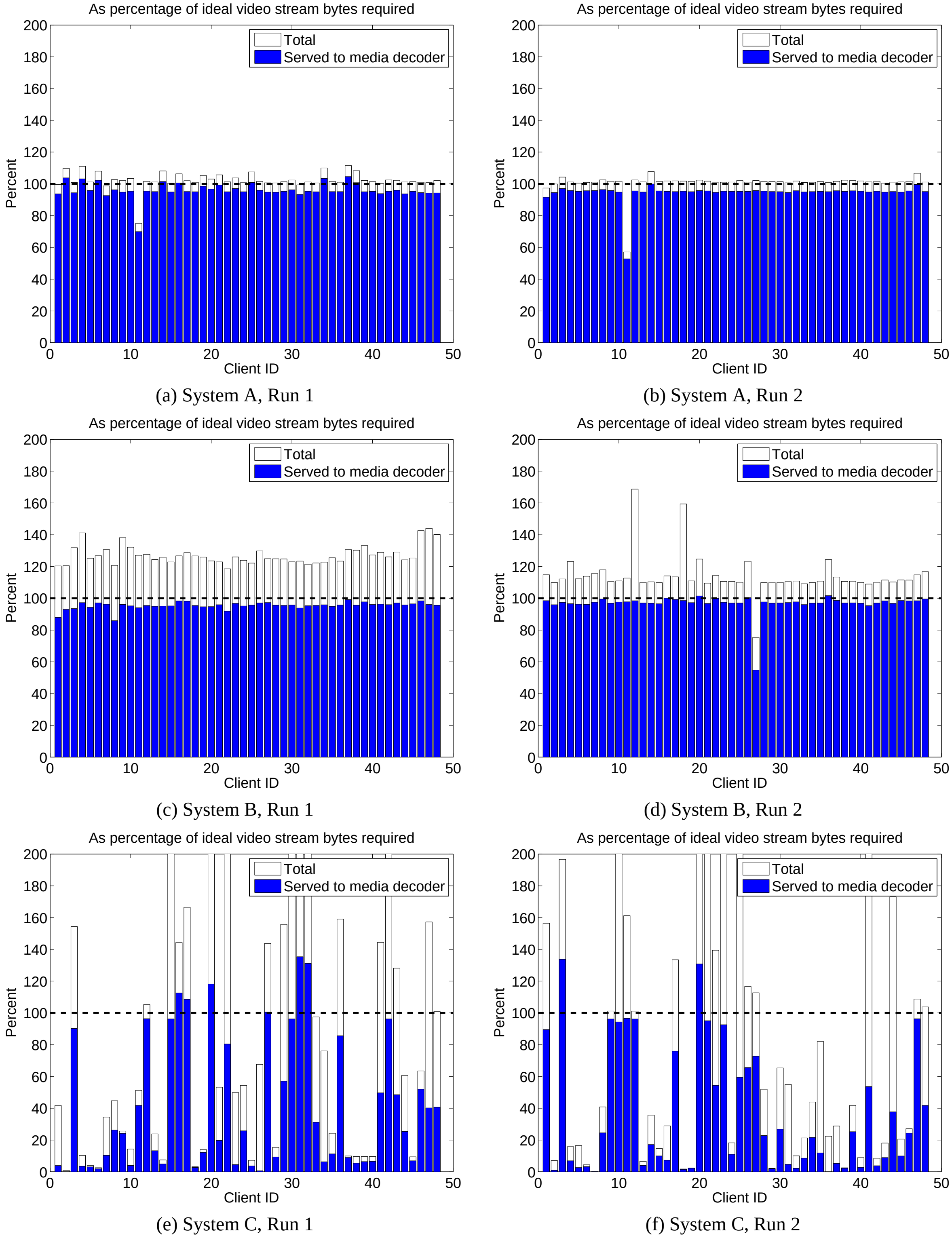
Figure 2 indicates that the constrained network (Run 1) was adequately provisioned to support Systems A and B, although System C performed poorly with only few peers getting the required amount of video stream. In addition, System A was significantly more efficient (approx. 6% overhead bandwidth) than System B (approx. 35% overhead for the constrained network). We conjecture that this significant difference may have arisen because of the overlay employing a tree-based protocol for System A as compared to the mesh-based overlay protocol of System B. The overhead of System B was approximately 20% for the unconstrained case (Run 2).

One of the most significant drawbacks of P2P streaming is the delay between the client starting up and the display of the required video stream. While some of this time is client dependent (speed of the client machine, etc.), a more important factor is the time taken by the P2P protocol to connect to the overlay network and negotiate stream transfer. In Figure 3, we measure the time taken from executing the P2P client's start command till the reception of the first video stream bytes on the client for Runs 1 and 2 respectively. Note that this time is the lower bound on the time a user waits before the video appears on her/his screen because of additional buffering/pre-roll delay of the media decoder. We emphasize that all the experiments in this work relate to live P2P streaming where, unlike video-on-demand scenarios, pre-emptive caching of content for instant playback is not a viable option.

System A took more time to connect to its P2P overlay than Systems B and C in general. However, a small start-up time did not translate into good performance for Solution C, as noted in Figure 2. We experimented with the required buffering/pre-roll delay for all three systems. This was done by slowly increasing the pre-roll delay till the point where no further improvement in video PSNR was observed. We noticed that a pre-roll delay of about 30 seconds was sufficient for System A, whereas for Systems B and C this value was close to 60 seconds.

Another interesting network measure is the comparison of the amount of the video stream downloaded from the server to that downloaded from other P2P clients. This measure is important for calculating the monetary cost for a content provider deploying P2P streaming systems because the fraction of bandwidth provided by the server is bought bandwidth. This measurement is depicted in Figure 4 for Runs 1 and 2 respectively.

System C reverted to almost a client-server model of downloading the bulk of the stream from the server. This may have lead to buffer overflows in the server and the consequent poor performance. On the other hand, despite the well-provisioned upload bandwidth in our test-bed, the intelligent algorithms in Systems A and B mostly used the P2P bandwidth even though the server had an uplink bandwidth of tens of Mbps. For example, for System B in Run 1, each peer downloaded 6.58 MB from the server and 110 MB from its peers on average. For Run 2, each peer downloaded 8.53 MB from the server and 85.87 MB from its peers on average. The greater server contribution in Run 2 is a factor in the average video quality of Run 2 being marginally better than that of Run 1 as the video quality PSNR drop measurements show.



(a) System A, Run 1

(b) System A, Run 2

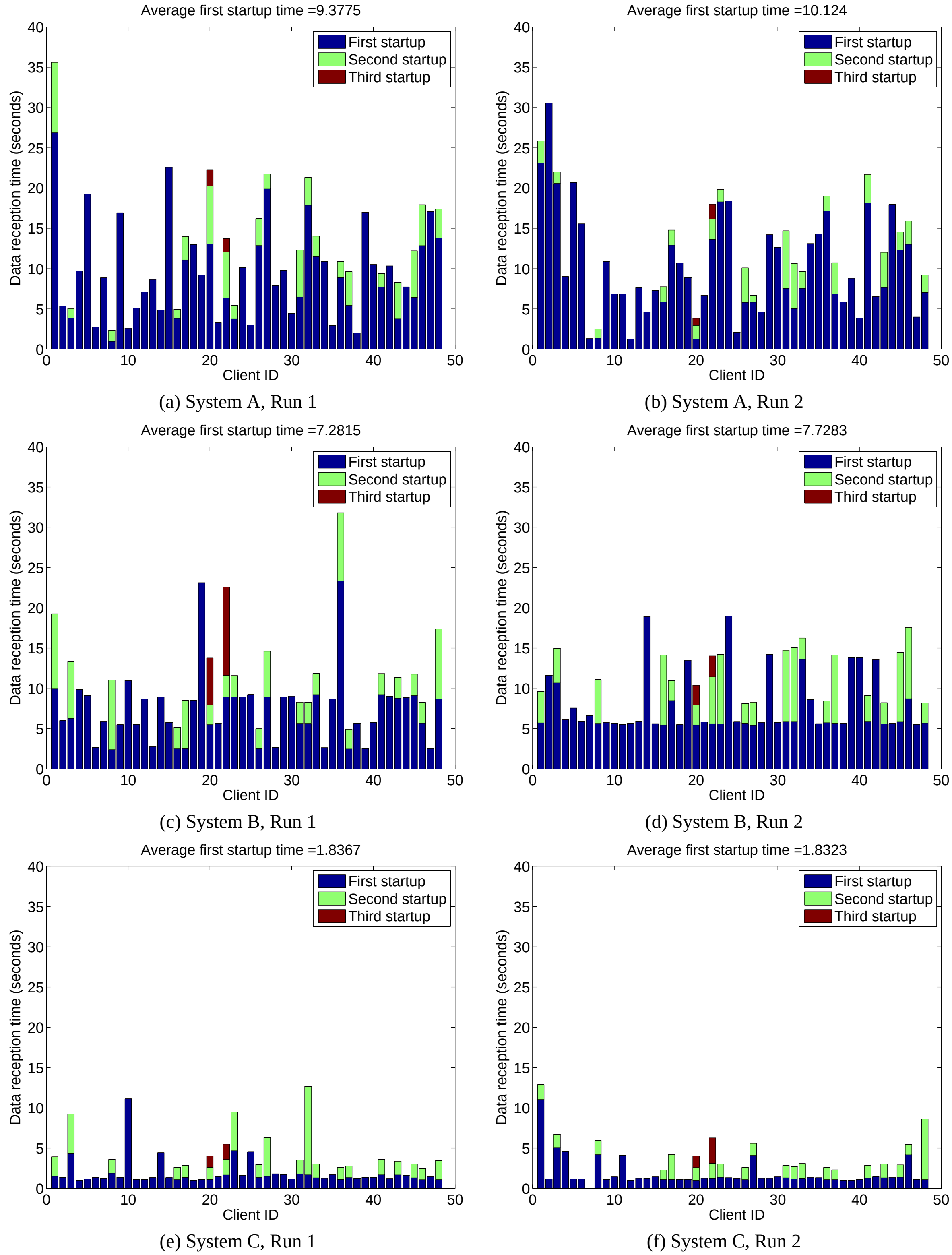
(c) System B, Run 1

(d) System B, Run 2

(e) System C, Run 1

(f) System C, Run 2

Figure 2: Protocol and duplicate overhead is indicated by the white bars. Filled bars indicate the percentage of required video stream served to the media decoder. Their shortfall (below 100%) introduces video playback quality degradation. Some bars exceed the scale employed.

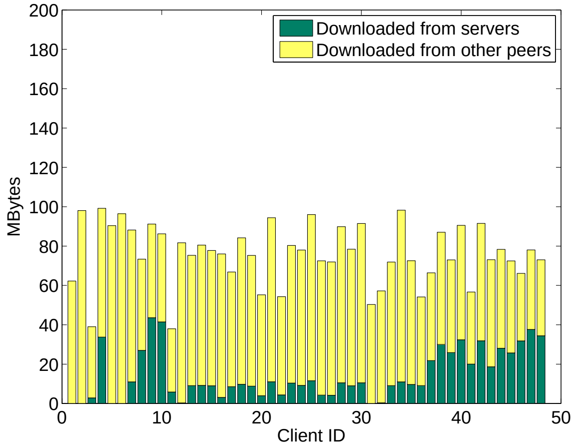


(a) System A, Run 1 (b) System A, Run 2

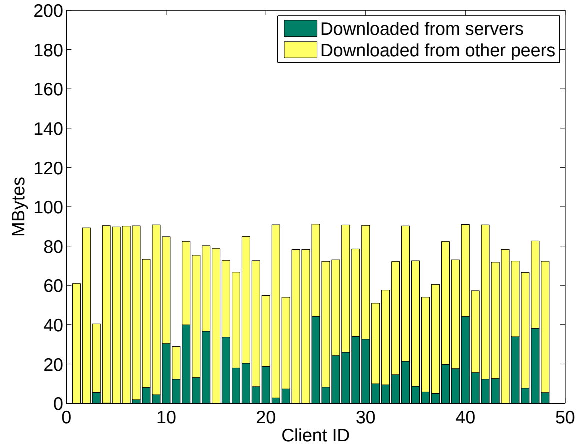
(c) System B, Run 1 (d) System B, Run 2

(e) System C, Run 1 (f) System C, Run 2

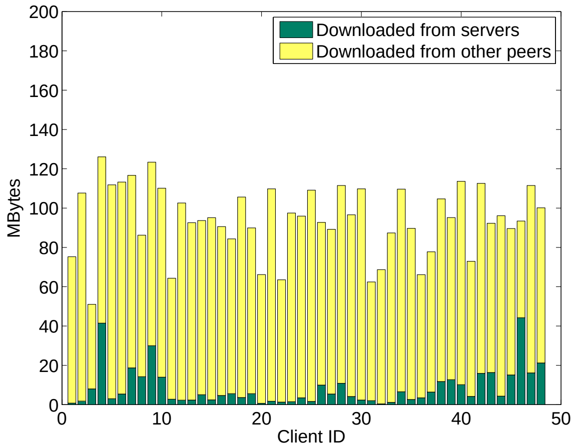
Figure 3: Startup delay: The time taken from executing the P2P client's startup command till when the first video stream bytes are received. Since the On-Off model is employed, note that some clients switch on and off multiple times.



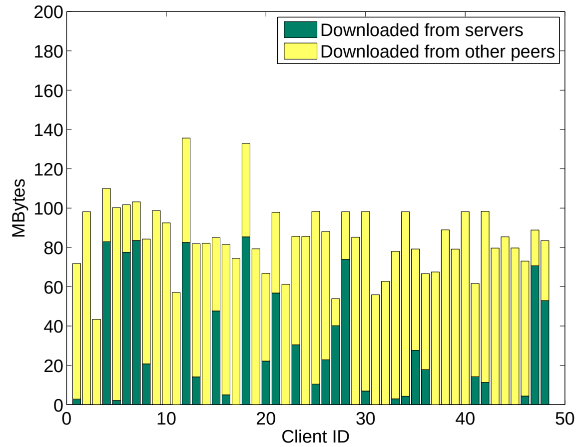
(a) System A, Run 1



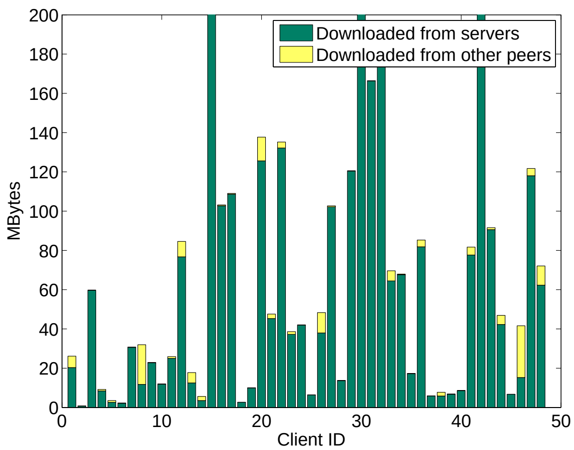
(b) System A, Run 2



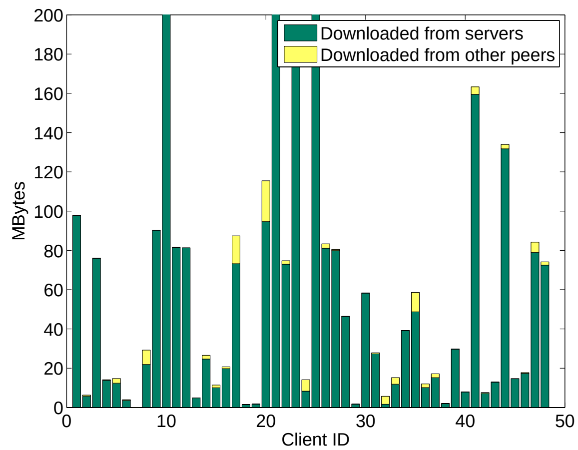
(c) System B, Run 1



(d) System B, Run 2

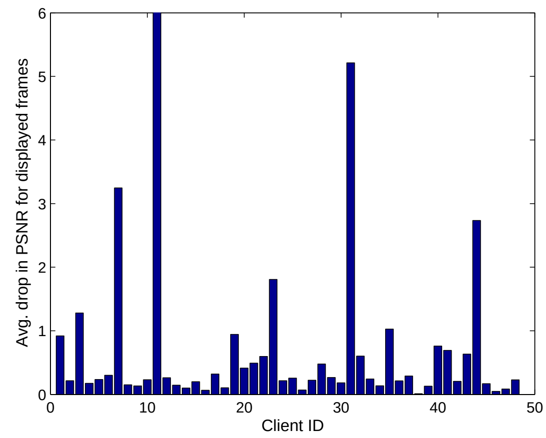


(e) System C, Run 1

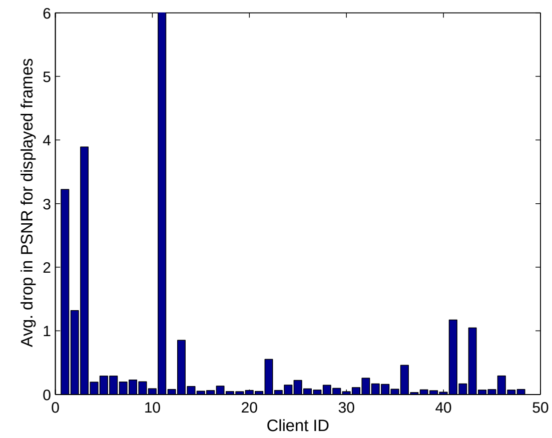


(f) System C, Run 2

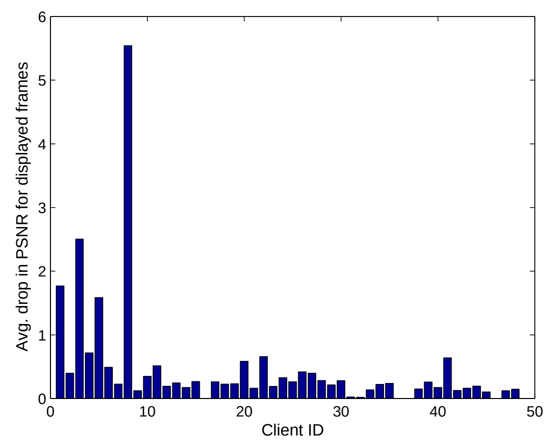
Figure 4: Comparison of bytes received from the server to bytes received from other peers. Since the On-Off model is employed, a peer might not need to download the entire video file. Some bars exceed the scale employed.



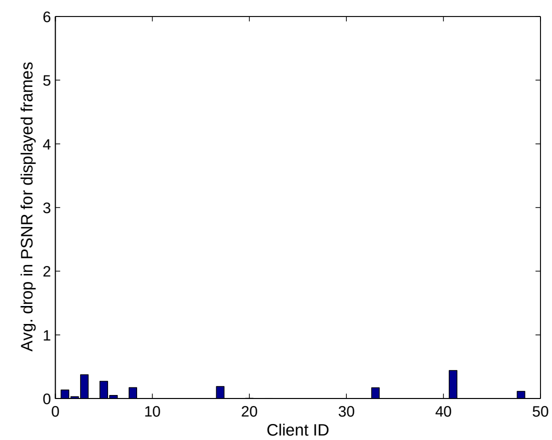
(a) System A, Run 1



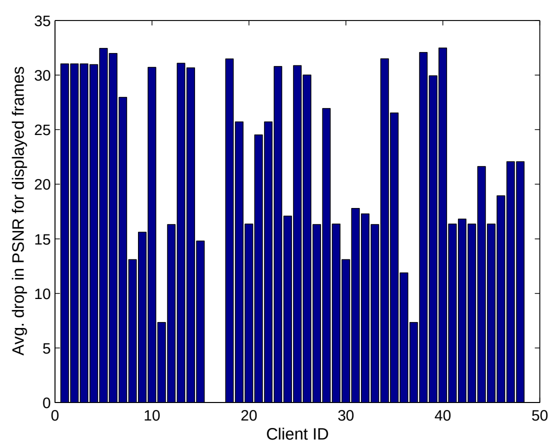
(b) System A, Run 2



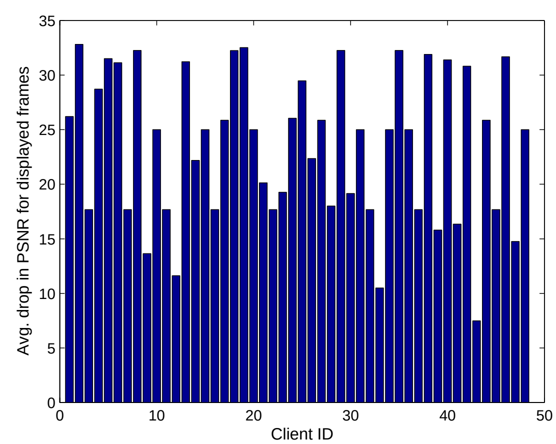
(c) System B, Run 1



(d) System B, Run 2

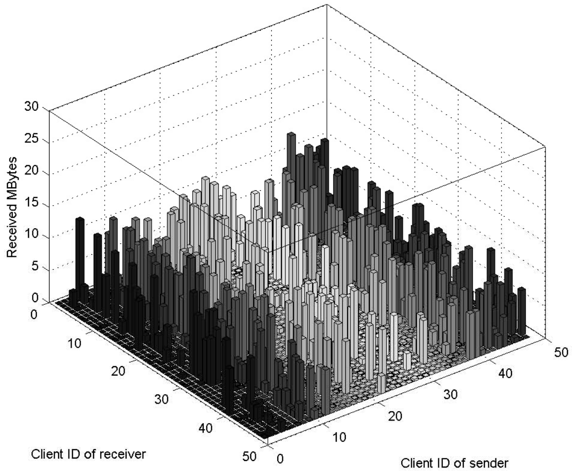


(e) System C, Run 1

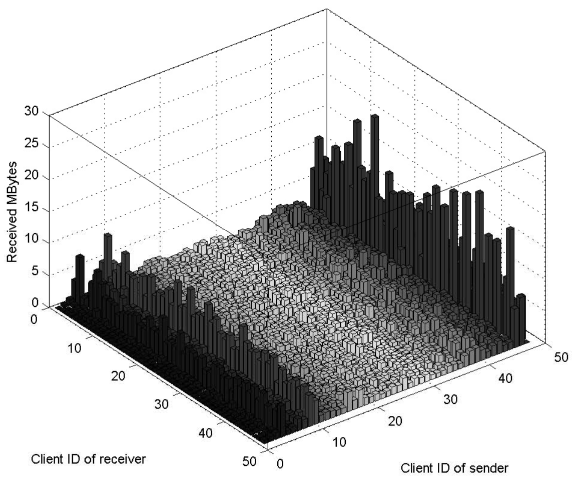


(f) System C, Run 2

Figure 5: Average drop in video quality over all the tested peers. Some bars exceed the scale employed.



(a) System A



(b) System B

Figure 6: Break-up of the total bytes received at each peer from every other peer during the P2P streaming session Run 1 with Systems A and B (some bars exceed the scale employed).

As noted before, Systems A and B have tree-based and mesh-based overlay architectures respectively. This distinction is responsible for the efficiency of System A, as discussed above. Figure 6 pictorially shows the transmit/receive foot print - number of bytes downloaded by each peer from every other peer in Run 1 - for both the systems. We do not report these measurements for System C because of its poor performance and overwhelming usage of the server bandwidth.

The underlying tree architecture of System A is immediately evident from the relatively sustained downloads on a peer from a few other peers. On the other hand, the mesh-based architecture of System B leads to smaller downloads on a peer from other peers. However, this also results in increased number of duplicates, hence reducing efficiency. This might be because of the fact that peers advertise for the data chunks that they have and then comply with requests from other peers to transmit these chunks. On the other hand, the redundancy of duplicate stream blocks may become useful when the network is highly dynamic or constrained.

As mentioned before, we fixed the pre-roll delay of the respective systems to a value beyond which no further video PSNR improvement was observed. We then evaluated the quality degradation in terms of the drop in the PSNR for all clients. This is shown in Fig. 5 for both test runs and averaged over the respective client's on-time. It is interesting to observe the correspondence between the shortfall depicted in Fig. 2 and the drop in PSNRs in Fig. 5. The first 8 clients have the DSL connection provided by an ISP, *i.e.*, these are not simulated in the data-center. In general, these clients experience the most degradation in quality. Client 8 experiences the worst degradation in quality due to it's under-performing ISP Internet connection.

6. CONCLUSION AND FUTURE WORK

P2P video streaming is now a mainstream Internet application with several commercial grade systems having large user bases. The traffic characteristics and network QoS requirements of P2P streaming systems are significantly different from other P2P applications given the real-time requirements for video streaming. We set up a controlled test-bed which emulates real-world network conditions through careful traffic shaping and then tested 3 commercially available P2P systems for different network conditions. The repeatability of our approach enables comparisons between different P2P streaming systems and/or testing the suitability of a particular network for a P2P streaming system.

Instead of limiting our measurements to packet loss and network usage only, we designed a novel P2P video quality measurement technique in order to obtain several received video quality measures such as video PSNR, channel changing times, etc. We also parsed our log data into relevant information like the efficiency and overhead of the tested P2P streaming systems, server-to-P2P bandwidth ratio and the P2P trasmit/receive footprint. These measures allowed us to make concrete statements about the effectiveness of each P2P streaming system.

Our results show that some of the tested P2P streaming systems have a significant overhead (up to 35% over the video stream size) and have an average start-up delay of under 11 seconds on our test-bed, although an additional video buffering time of at least 30 seconds is needed to combat the effect of variations in the packet arrival times on the video playback. We also observed that the P2P systems are robust to peer churn and intelligently use P2P bandwidth instead of simply resorting to downloading the video stream from our over-provisioned server. Finally, we report that there are substantial differences between P2P streaming systems' performance based upon their underlying implementation and the choice of protocols.

Acknowledgment

The authors thank the P2P system providers for providing the P2P streaming system for academic study. The P2P TV-ID project at T-Labs and T-Systems provided test-bed support. They also thank Anja Feldmann for useful discussions.

7. REFERENCES

[1] NISTNet. http://dsd.lbl.gov/NCS/netest/.
[2] T-Systems International IP-Network Statistics Web Site. http://www.t-statistics.com/.
[3] S. Ali, A. Mathur, and H. Zhang. Measurement of commercial peer-to-peer live video streaming.
[4] P. Baccichet, J. Noh, E. Setton, and B. Girod. Content-aware p2p video streaming with low latency. *In Proc. of IEEE*

International Conference on Multimedia and Expo, ICME, Jul 2007.

[5] Carnegie Mellon University. End System Multicast. http://www.cs.cmu.edu/ esm.

[6] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: high-bandwidth multicast in cooperative environments. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, 2003.

[7] Y. Chawathe. Scattercast: an adaptable broadcast distribution framework. *Multimedia Syst.*, 9(1), 2003.

[8] Coolstreaming. Coolstreaming. http://www.coolstreaming.us/.

[9] G. A. Fowler and S. McBride. Newest Export from China: Pirated Pay TV. *Wall Street J.*, Sept. 2 2005.

[10] Gridmedia. Gridmedia. http://www.gridmedia.com.cn/.

[11] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross. A Measurement Study of a Large Scale P2P IPTV System. Technical report, Department of Computer and Information Science, Polytechnic University, New York, USA, 2006.

[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 Proc. of IPTV Workshop, 15th International World Wide Web Conference*, May. 2006.

[13] Hobbit. Hobbit Monitoring Tool. http://hobbitmon.sourceforge.net/docs/man7/hobbit.7.html.

[14] ITU-T and ISO/IEC JTC 1. *Advanced Video Coding for Generic Audiovisual services, ITU-T Recommendation H.264 - ISO/IEC 14496-10(AVC)*, 2003.

[15] J. Klaue, B. Rathke, and A. Wolisz. EvalVid - A framework for video transmission and quality evaluation. *In Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Urbana, IL*, pages 255–272, Sept. 2003.

[16] E. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *IEEE Communications Surveys and Tutorials*, 7(2):72–93, Oct. 2005.

[17] N. Magharei, R. Rejaie, and Y. Guo. Mesh or Multiple-Tree: A Comparative Study of Live Peer-to-Peer Streaming Approaches. *In Proc. IEEE INFOCOM*, May. 2007.

[18] J. Navratil and R. L. Cottrell. Abing. http://www-iepm.slac.stanford.edu/tools/abing/.

[19] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai. Distributing streaming media content using cooperative networking. In *ACM NOSSDAV*, 2002.

[20] Planet-Lab. Planet-Lab. http://www.planet-lab.org/.

[21] PPLive. PPLive. http://www.pplive.com/.

[22] P. Seeling, F. Fitzek, and M. Reisslein. *Video Traces for Network Performance Evaluation: A Comprehensive Overview and Guide on Video Traces and Their Utilization in Networking Research*. Springer Verlag, 2007.

[23] P. Seeling, M. Reisslein, and B. Kulapala. Network performance evaluation with frame size and quality traces of single-layer and two-layer video: A tutorial. *IEEE Communications Surveys and Tutorials*, 6(3):58–78, Third Quarter 2004.

[24] A. Sentinelli, G. Marfia, M. Gerla, and L. Kleinrock. Will IPTV ride the Peer-to-Peer stream? *IEEE Communications Magazine*, 45(6):86–92, June 2007.

[25] E. Setton, P. Baccichet, and B. Girod. Peer-to-peer live multicast: a video perspective. *Proceedings of the IEEE*, 96(1):25–38, Jan. 2008.

[26] T. Silverston and O. Fourmaux. P2P IPTV Measurement: A Comparison Study. http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0610133, 2006.

[27] SopCast. Sopcast. http://www.sopcast.org/.

[28] Y. Tang, J. Luo, Q. Zhang, M. Zhang, and S. Yang. Deploying p2p networks for large-scale live video-streaming service. *IEEE Communications Magazine*, 45(6):100–106, June 2007.

[29] M. Zhang, L. Zhao, Y. Tang, J. Luo, and S. Yang. Large-Scale Live Media Streaming over Peer-to-Peer Networks through the Global Internet. *In Proc. of ACM Multimedia 2005, Workshop on Advances in Peer-to-Peer Multimedia Streaming*, Nov. 2005.

[30] X. Zhang, J. Liu, and B.Li. On Large Scale Peer-to-Peer Live Video Distribution: Coolstreaming and its Preliminary Experimental Results. *In Proc. of IEEE International Workshop on Multimedia Signal Processing, MMSP*, Oct. 2005.

[31] X. Zhang, J. Liuy, B. Liz, and T. Yum. Coolstreaming/DONet: A Data-Driven Overlay network for efficient Live Media Streaming. *In Proc. of IEEE INFOCOM*, March 2005.