VIDEO QUALITY ASSESSMENT AND COMPARATIVE EVALUATION OF PEER-TO-PEER VIDEO STREAMING SYSTEMS

Aditya Mavlankar, Pierpaolo Baccichet, Bernd Girod

Department of Electrical Engineering
Stanford University
Stanford, CA 94305, USA

Sachin Agarwal, Jatinder Pal Singh

Deutsche Telekom A.G., Laboratories
Ernst-Reuter-Platz 7
10587 Berlin, Germany

ABSTRACT

We design a test methodology to analyze in detail the video quality received at each peer in a peer-to-peer (P2P) video streaming system. The metrics that we employ at each peer include video PSNR, statistical analysis of frame-freeze events, the amount of time to wait before video playback starts, nature of the data-paths established to serve the peer, protocol overhead and duplicate data received. These metrics are estimated by analyzing the packet reception times at each peer and utilizing information about the original uncompressed video as well as the encoded video. We use this framework to compare the performance of three P2P video streaming systems by deploying them on our controlled traffic-shaped network test-bed. We can emulate the same network conditions and peer behavior for testing different systems and ensure that the experiments are repeatable. These measurements highlight the differences between systems based upon their underlying implementation, overlay architecture, and choice of protocols. This measurement study helps to gauge the performance of currently available P2P video streaming systems and points out desirable performance improvements.

Index Terms— peer-to-peer video streaming, video traces, overlay architectures

1. INTRODUCTION AND RELATED WORK

In the recent years, numerous academic and commercial Internet peer-to-peer (P2P) streaming systems have become available. Some systems already have large installed user bases for both live and on-demand video streaming, for example [1–4]. A survey and comparison of the different approaches and protocols for P2P streaming systems can be found in [5,6]. To gauge their strengths and weaknesses, a methodology for comparing different implementations is required. We propose such a methodology that allows head-to-head comparison using several relevant parameters; at each receiving peer, these include video PSNR, statistical analysis of frame-freeze events, the amount of time to wait before video playback starts, nature of the data-paths established to serve the peer, protocol overhead and duplicate data received, etc. We also build a controlled network test-bed according to various measurements on the Internet. Finally, we deploy three commercial-grade systems on the test-bed and use the proposed methodology for analyzing their performance. We contrast our work against recent studies, for example [7–11]. Firstly, these studies do not focus on received video quality measures such as PSNR and video startup times and are limited to an analysis of the networking characteristics like bandwidth usage, packet loss, etc. For example, [12] is a recent measurement study on SopCast that reports an extensive list of metrics but does not include video PSNR. Secondly, most of these measurement studies are done on the basis of logs from real sessions on the Internet and this entails little possibility of fair comparison owing to differences in infrastructure, network conditions, video characteristics and peer behavior. On our controlled test-bed, we can carefully select these parameters and use the same settings for all tested systems. We recently reported the performance of the three systems based on some networking-centric metrics in [13]; it complements the analysis in this paper.

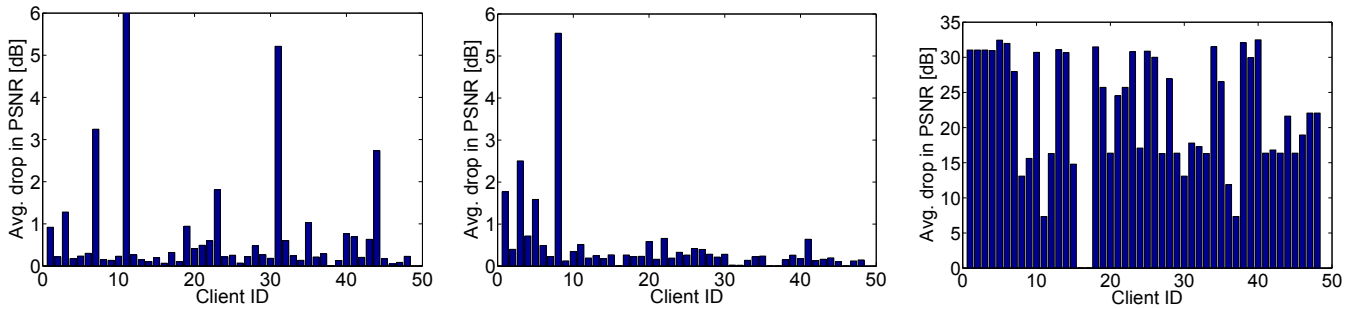
2. TEST-BED

We set up a controlled IP network by simulating real-world network conditions using the NISTNet tool [14] for traffic shaping. The conditions are based on prior measurements between real hosts in Berlin (Germany), Stanford (USA) and Munich (Germany). We simulated 48 clients/peers in all; 23 in Berlin, 22 in Stanford and 3 in Munich. The upload bandwidth distribution among these 48 clients, also controlled using NISTNet, is given by $\{3072 \times 3, 2048 \times 2, 1024 \times 12, 576 \times 27, 192 \times 2, 128 \times 2\}$, in $b \times n$ notation meaning n clients with an upload bandwidth of b kbps. The hardware emulating 40 of the clients is inside a Deutsche Telekom data-center in Erfurt, Germany. The remaining 8 clients are hosts in Berlin with DSL Internet connections; their distribution is $\{1024 \times 2, 576 \times 2, 192 \times 2, 128 \times 2\}$. These 8 clients, denoted by client IDs 1 through 8 in the next section, are not traffic-shaped through NISTNet as they are limited by real DSL connections. The P2P server is hosted by another Deutsche Telekom data-center in Berlin.

We employed the following peer churn model. During each 6 minute time slot, a peer is on or off with probabilities 0.9 and 0.1 respectively. Also, a peer can switch off for the rest of the run during any time slot with probability 0.05. In the last 5 minutes of the run, a peer is off with probability 0.5. This emulates the scenario when peers rapidly depart the P2P overlay at the end of the session.

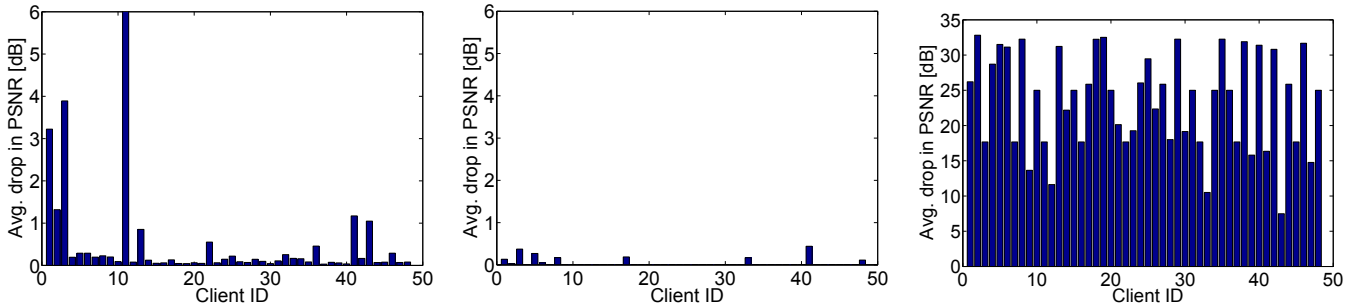
3. EXPERIMENTS

The three tested systems are in an advanced stage of development and have been successfully deployed on the Internet. In order to protect commercial interests, we refer to them as System A, System B and System C. For the P2P overlay, System A builds multiple multicast trees and adopts the so-called push approach. Systems B and C are mesh-based and follow the so-called data-driven approach or pull approach. We report results for two test-runs for each system. The NISTNet tool performed traffic shaping in Run 1, whereas Run 2 was with NISTNet disabled and hence the underlying physical network characteristics were applicable. The same realization of the statistical On-Off model was used in all runs for emulating peer



(a) System A, Run 1 (b) System B, Run 1 (c) System C, Run 1

Fig. 1. Average drop in video quality for the displayed frames. The drop is shown for all 48 clients for Run 1.



(a) System A, Run 2 (b) System B, Run 2 (c) System C, Run 2

Fig. 2. Average drop in video quality for the displayed frames. The drop is shown for all 48 clients for Run 2.

churn. Please note that for each run, we performed multiple trials and observed low variance in the results. Hence, we pick a single trial for reporting the results for each run, since a statistical average of quality across multiple trials would hide some of the adverse effects that users observe in video playback.

We encoded 30 minutes of the classic movie *La Dolce Vita* (Fellini, 1960), using a state-of-the-art H.264/AVC [15] video codec with accurate rate-control to generate a 400 kbit/s CBR bit-stream. The spatial resolution is 352×240 pixels, the frame-rate is 24 fps and the average PSNR of the encoded sequence is approximately 42 dB. We omit the audio stream in this work. An intra-coded picture (I) is inserted every second and the number of consecutive bi-directionally predicted (B) frames is two. The uni-directionally 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.

Due to network packet loss, peer churn, etc., some parts of the bit-stream never arrive or arrive too late for playout and this affects the quality of the displayed video. 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 appears frozen. The loss of a large portion of contiguous data causes the loss of several consecutive frames and leads to a long frame freeze. We assume that Accelerated Retroactive Decoding (ARD) [16] is possible; i.e., even though a frame arrives late for its own display, it can still be used for decoding a future frame that depends on it.

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.

We obtained slightly modified versions of the systems that accurately log the packet arrival times. 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. The waiting time from connection initiation until the playout is called pre-roll delay or buffering time. It should be noted that although a given system might get the first few data-bytes with low delay, it could still require long buffering time to sustain good quality. 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. Similar techniques for translating packet loss and arrival times into

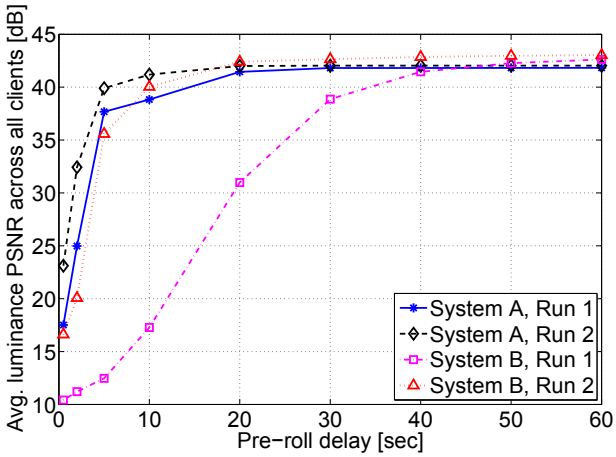


Fig. 3. Average quality across all peers as a function of the pre-roll delay.

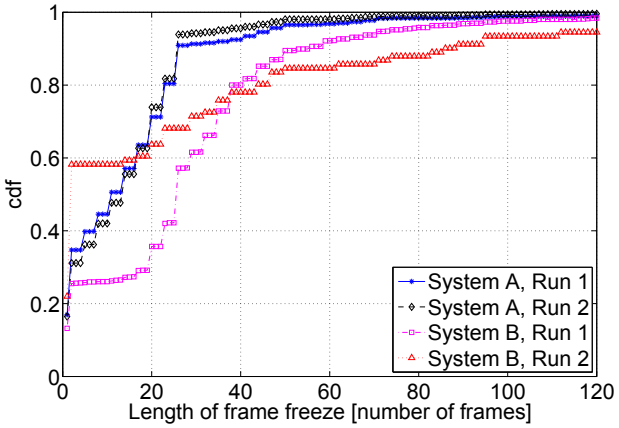


Fig. 4. Cumulative density function (cdf) for lengths of frame freezes. The cdf is computed over all 48 clients.

video PSNR have been proposed in [17, 18]. Reference [19] uses a similar technique to analyze video quality in experiments with peer-to-peer video streaming over the Planet-Lab test-bed. However, the setup in [19] was limited to high-speed university connections with no peer churn.

For Systems A and B, Fig. 3 shows the video quality averaged across all peers as a function of the pre-roll delay. The increase of pre-roll delay beyond a certain value does not yield any further improvement. 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. In the rest of the paper, we report other results obtained by assuming these values of the pre-roll delay for the respective systems.

For all three systems, Fig. 1 shows the average drop in PSNR for all 48 clients for Run 1. Figure 2 shows the average drop in PSNR for Run 2. System B shows noticeable improvement after removing the bandwidth constraints compared to Run 1, whereas System A shows marginal improvement. We conjecture that System C had problems reaching clients behind a Network Address Translator (NAT). The peers that get most of the data directly from the server could receive good quality and most other peers facing the NAT resolution problem experienced a dramatic loss of quality. In this paper, we omit some results for System C due to this technical shortcoming.

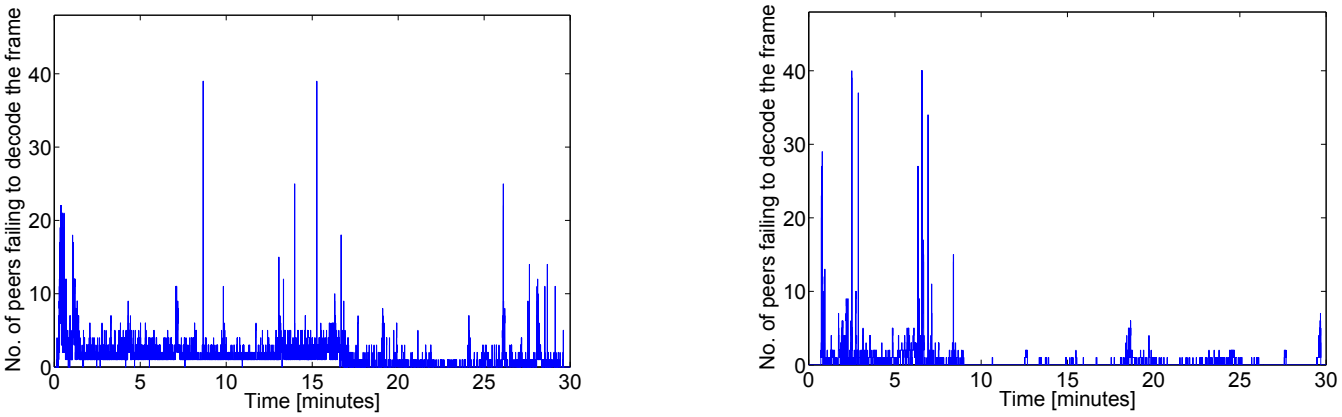
We now present the statistics for frame freezes experienced by the clients. For Systems A and B, Fig. 4 shows the cumulative density function (cdf) of the lengths of frame freezes over all the clients. The percentage of frames frozen out of the total frames to be displayed for all clients is about 4.21%, 3.02%, 2.02% and 0.15% for System A's Run 1 and Run 2 and System B's Run 1 and Run 2 respectively. The average number of distinct freeze events per client is about 64, 40, 23 and 2 respectively. It can be seen from the cdf that frame freezes longer than 100 frames are rare but do occur at times, especially for System B. System A has a better cdf overall. We conjecture that this is because, in System A, the sending peer adopts content-aware prioritization among the packets in its transmission buffer. The transmission buffer also holds outgoing packets that are responses to retransmission requests from the receiving peer. This algorithm is aware of the coding dependencies and serves to reduce the duration of a frame freeze event.

We also measured the percentage of redundancy in the delivered stream and the upload bandwidth used at the server. We define a video stream as the bytes fed to the media decoder by the P2P client running on a peer. These data are stripped of any protocol control packets, headers, etc. and only comprise of video data; in our case, this is the H.264/AVC stream itself or the same wrapped in its appropriate media container such as Microsoft's ASF (Advanced Systems Format). We monitor the bytes received at the network interface of each client and calculate the amount of redundancy as the bytes received in excess of the video stream delivered to the media decoder. Next, we report this redundancy as a percentage of the required video stream bytes.

Due to its tree-based approach, System A creates approximately 6% redundancy compared to System B's approximately 35% redundancy. We also observed that when the traffic shaping was disabled in Run 2, the redundancy for System B reduced to approximately 20%, although it did not change for System A. On further analysis of the data-paths at every peer, we found that peers in System A have sustained downloads from a few other peers. On the other hand, the mesh-based architecture of System B leads to smaller downloads from individual peers but the number of peers delivering the content to any given peer is higher than in System A. This is because peers advertise for the data chunks that they have and then comply with requests from other peers to transmit these chunks. This makes it harder for peers to co-ordinate these downloads and eliminate duplicates. While the mesh-based approach of System B leads to more duplicates, it could lend required robustness in the case of high peer churn.

Now we compare the amount of data downloaded from the server versus the amount downloaded from other peers. This comparison is made for the total received data at the peer including protocol overhead. It should be noted that despite the uplink bandwidth of the server being tens of Mbps, both System A and System B receive less than 10% of their data from the server directly. We observed that System B uses up slightly more bandwidth from the server in Run 2 compared to Run 1. This probably eases the coordination for data distribution and may explain the lesser overhead in Run 2 for System B compared to Run 1.

Finally, Fig. 5 shows the profile of the number of peers that fail to decode each video frame for Systems A and B. There are a few instances when almost 90% of peers fail to decode a particular video frame. Also, towards the end of the session when peers depart the overlay rapidly, the number of failures increases slightly.



(a) System A, Run 1

(b) System B, Run 1

Fig. 5. Profile of number of peers unable to decode the video frame.

4. CONCLUSIONS

Our methodology for testing the performance of P2P video streaming systems allows to measure several important quality metrics beyond packet loss and network usage; the metrics that we employ include required buffering time, video PSNR, frame freeze statistics, number of peers failing to decode the video frame, etc.

We employ our methodology to test three commercial-grade P2P video streaming systems on our controlled test-bed. The test-bed has an advantage that the test conditions can be chosen by analyzing real-world conditions and the experiments are repeatable. We find that the state-of-the-art systems are quite efficient in reducing the load on the server by using the P2P bandwidth. Although in general, the losses are reasonably low, there are instances when the display freezes for more than 100 frames and also the required buffering time for all tested systems was of the order of tens of seconds. This suggests room for improvement in peer-to-peer video streaming. Our experiments reveal that there are substantial differences between different systems based upon their underlying implementation and choice of protocols. In particular, we observed that the tested system using a tree-based push approach outperforms the tested mesh-based system in terms of pre-roll delay and generated redundancy by a considerable margin.

5. REFERENCES

[1] "SopCast," online: http://www.sopcast.org/.

[2] "PPLive," online: http://www.pplive.com/.

[3] "Coolstreaming," online: http://www.coolstreaming.us/.

[4] "GridMedia," online: http://www.gridmedia.com.cn/.

[5] 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*, vol. 7, no. 2, pp. 72–93, Oct. 2005.

[6] 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.

[7] Shahzad Ali, Anket Mathur, and Hui Zhang, "Measurement of commercial peer-to-peer live video streaming," online: http://www.citeseer.ist.psu.edu/ali06measurement.html.

[8] 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 Intl. World Wide Web Conf.*, May 2006.

[9] 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 Intl. Workshop on Multimedia Signal Processing, MMSP*, Oct. 2005.

[10] 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.

[11] Y. Tang, J.G. Luo, Q. Zhang, M. Zhang, and S.Q. Yang, "Deploying P2P networks for large-scale live video-streaming service," *IEEE Communications Magazine*, vol. 45, no. 6, pp. 100–106, June 2007.

[12] A. Sentinelli, G. Marfia, M. Gerla, and L. Kleinrock, "Will IPTV ride the Peer-to-Peer stream?," *IEEE Communications Magazine*, vol. 45, no. 6, pp. 86–92, June 2007.

[13] S. Agarwal, J. P. Singh, A. Mavlankar, P. Baccichet, and B. Girod, "Performance of P2P live video streaming systems on a controlled test-bed," *In Proc. of 4th Intl. Conf. on Testbeds and Research Infrastructures for the Development of Networks & Communities, TRIDENTCOM, Innsbruck, Austria*, Mar. 2008.

[14] "NISTNet," online: http://dsd.lbl.gov/NCS/netest/.

[15] 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.

[16] M. Kalman, P. Ramanathan, and B. Girod, "Rate-distortion optimized video streaming with multiple deadlines," *In Proc. of IEEE Intl. Conf. on Image Processing, ICIP*, Sept. 2003.

[17] 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.

[18] J. Klaue, B. Rathke, and A. Wolisz, "EvalVid - A framework for video transmission and quality evaluation," *In Proc. of the 13th Intl. Conf. on Modeling Techniques and Tools for Computer Performance Evaluation*, pp. 255–272, Sept. 2003.

[19] P. Baccichet, J. Noh, E. Setton, and B. Girod, "Content-aware P2P video streaming with low latency," *In Proc. of IEEE Intl. Conf. on Multimedia and Expo, ICME*, Jul 2007.