Self-organizing QoE and Bandwidth Control in P2P VoD Streaming

Simon Oechsner, Thomas Zinner, Tobias Hoßfeld
University of Würzburg, Institute of Computer Science, Germany
Email: {oechsner|zinner|hossfeld}@informatik.uni-wuerzburg.de

I. Introduction

In recent studies on the traffic distribution in the Internet, e.g., [1], video streaming and especially Video-on-Demand (VoD) such as offered by YouTube are shown to gain more and more popularity. The traffic share of this application class grows accordingly. P2P overlays are prime candidates for efficiently streaming video content. Clients like Tribler [2] already support VoD services based on well-known mechanisms from traditional file-sharing.

In parallel to this development, the number and variety of end user devices that can play back a video in acceptable quality has grown. A video may be streamed to a mobile device connected via UMTS, a laptop in a WLAN or to a LCD TV with a broadband connection to the Internet. However, due to their different capabilities and access bandwidths, there is no single ideal video stream for all devices.

To be able to support heterogeneous end user devices, a content provider might offer the same video in different levels of quality, resulting in one video stream per version. A more efficient solution is the recent addition of Scalable Video Codecs (SVC) [3] to the popular H.264 coding standard [4]. This codec allows for a separation of a single source video file into layers which can theoretically be extracted from the stream with no additional coding effort.

We present a P2P VoD system based on Tribler that supports a SVC video and is able to adapt to different network conditions.

II. Scalable Video Codecs

The video codec H.264/SVC [5] [3] is based on H.264/AVC, a video codec used widely in the Internet, for instance by video platforms (e.g., YouTube, GoogleVideo) or video streaming applications (e.g., Zattoo). H.264/AVC is a so called single-layer codec, which means that different video files have to be provided to support different end user devices. The Scalable Video Coding (SVC) extension of H.264/AVC enables the encoding of a video file at different qualities within the same layered bit stream. This includes besides different resolutions also different frequencies (frames displayed per second) and different qualities w.r.t. Signal-to-Noise Ratio (SNR). These can be considered as a special case of spatial scalability with identical picture size for base and enhancement layers. These three dimensions of enhancements are denoted as spatial, temporal and quality scalability.

Figure 1 gives an example of different possible scalabilities for a video file. The scalable video file can be watched in three different temporal resolutions (15Hz, 30Hz, 60Hz), three different spatial resolutions (CIF, SD, HD) and three different quality resolutions (Q0, Q1, Q2). The left bottom "subcube", CIF resolution with 15 Hz and quality Q0, is the base layer which is necessary to play the video file. Based on this layer different enhancement layers permit a better video experience with a higher resolution, better SNR or higher frame rate, respectively. The more subcubes along any of the three axes are available the higher the quality in this respect is. If all subcubes are available the video can be played back in highest overall quality. If all subcubes within quality Q0 are available, the video can be played back in HD-resolution with 60 HZ, but only with a low SNR quality.

III. Architecture Description

The P2P VoD architecture we evaluate is based on Tribler, which in turn is a video streaming BitTorrent adaptation.

We only consider the temporal scalability of a video in SVC. Basically, this means that the frames of the complete video are separated into layers, with each additional layer doubling the frame rate of the video. Since the enhancement layers are referencing all layers below them, they cannot be played out without these layers. Only the base layer contains frames that exclusively reference frames in the same layer, meaning that this layer can be played out by itself.

Our aim is to download chunks so that a video quality is attained that can be supported by the network capacity. Therefore, we prioritize lower layers over higher ones, while still keeping the set separation and rough in-order download strategy of Tribler (cf. Fig. 2). In the high-priority set, we only download the base layer of the video to make sure we can always play back the video and to avoid stalling times. This means we never download the enhancement layers from the beginning of the video, which is a minor limitation of our architecture and could be circumvented by a longer buffering time. Beginning with the second set, we first download chunks of the base layer according to rarest first. If all of these are already downloaded or no chunk from that layer can be selected, we start downloading chunks from enhancement layer 1, also choosing the rarest chunk first. In general, we only download chunks from a higher enhancement layer if all chunks of the lower layers are locally available or cannot be selected.

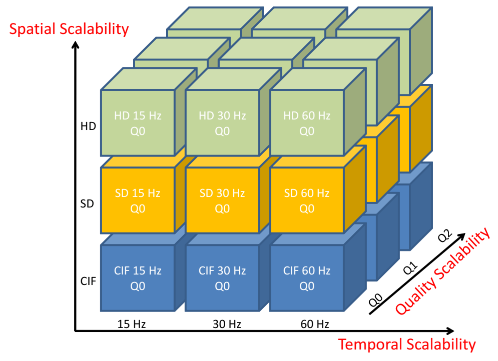

Fig. 1. SVC Cube, illustrating the possible scalability dimensions for a video file

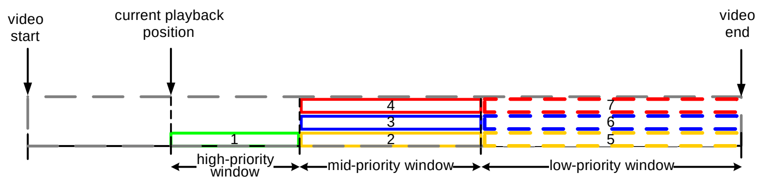

Fig. 2. Adapted chunk selection priority sets

Finally, we have the same selection process in the low-priority set, which is only considered if all chunks from the other two sets are downloaded or unavailable from other peers.

IV. Numerical Results

Among other scenarios, we consider in our evaluation scenarios where we have two classes of peers with different upload bandwidths. These are based on realistic upload access speeds for DSL1000 and DSL2000 connections, and are set to 128kbps and 192kbps, respectively. We do not only vary the load by changing the share of the different peer classes, but we also let half of the content provider streaming servers fail simultaneously after half of the steady-state simulation time. Thus, we want to see how the chunk selection process reacts to the different load conditions, without any parameters being changed. Fig. 3 and 4 show the results. The x-axis denotes the share of DSL1000 peers to DSL2000 peers as, e.g., 10/90 if 10% of the peers are DSL1000 peers and the rest DSL2000 peers.

We see that the peers with a higher upload capacity have a better QoE on average than their counterparts with less capacity. They can play out more layers more often, and experience a much shorter mean stalling time. We also can observe the effect of changing the peer set composition to include a higher share of peers with less upload capacity. Since this effectively means reducing the total upload capacity available to the system while keeping the download demand constant, a reduction in the QoE is to be expected and can be observed.

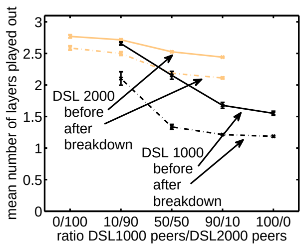

Fig. 3. Mean number of layers played out for different peer set compositions, before and after server breakdown

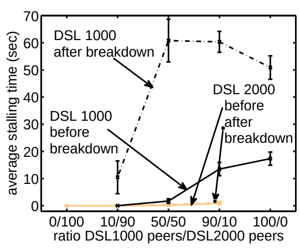

Fig. 4. Mean stalling times for different peer set compositions, before and after breakdown

However, the DSL1000 peers suffer much more from this reduction of resources than the DSL2000 peers. While the DSL2000 peers experience a slight decrease in quality only, the average number of layers played out drops noticeably for the DSL1000 peers. Also, the mean stalling times increase drastically. This effect is exacerbated after half of the servers have failed, since then even less upload capacity is available in the network.

In general, however, the chunk selection strategy adapted for SVC copes with the node failures and according load increase during the swarm lifetime without any parameters needed to be adapted to the new situation.

V. Conclusion

The presented P2P VoD architecture supports streaming a SVC video. We extended the chunk selection strategy in a straightforward way to accommodate the layered structure of the video file. Since Tribler is a deployed and used application, this means our approach is realistic and can be implemented in a live P2P VoD system.

Our performance evaluation shows that the proposed strategy is able to adapt to the system load and peer access capabilities without having to measure network conditions or relying on feedback from the video player software. Additionally, it conserves and even reinforces incentives generated by the G2G peer selection to provide resources to the overlay.

Acknowledgments

This work has been conducted in the project G-Lab, funded by the German Ministry of Educations and Research (Förderkennzeichen 01 BK 0800, G-Lab). The authors alone are responsible for the content of the paper.

References

[1] Ipoque - Internet Study 2008/2009, “http://www.ipoque.com/,” 2009.

[2] Garbacki P. Wang J. Bakker A. Yang J. Iosup A. Epema D. Reinders M. Van Steen M. Pouwelse, J. and H. Sips, “Tribler: A social-based peer-to-peer system,” in *In The 5th International Workshop on Peer-to-Peer Systems (IPTPS’06)*, 2006, pp. 1–6.

[3] ITU-T Rec. & ISO/IEC 14496-10 AVC, , “Advanced Video Coding for Generic Audiovisual Services,” 2007.

[4] Marpe, D. and Wiegand, T. and Sullivan, GJ, “The H. 264/MPEG4 advanced video coding standard and its applications,” *IEEE Communications Magazine*, vol. 44, no. 8, pp. 134–143, 2006.

[5] Marpe D. Wiegand T. Schwarz, H., “Overview of the scalable video coding extension of the h.264/avc standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1129, 2007.