

LiveShift: Peer-to-peer Live Streaming with Distributed Time-Shifting

Fabio Victora Hecht¹, Thomas Bocek¹, Cristian Morariu¹, David Hausheer¹, Burkhard Stiller^{1,2}

¹ University of Zurich, Institute of Informatics (IFI), Zurich, Switzerland

² ETH Zurich, Computer Engineering and Networks Lab (TIK), Zurich, Switzerland
{hecht,bocek,morariu,hausheer,stiller}@ifi.uzh.ch

Abstract

The increasing assortment of devices with IP connectivity contributes to the high popularity of video sharing over the Internet. High traffic generated by such applications at the source can be better distributed using a peer-to-peer overlay, since every user forwards information to other users. Current implementations target either live or on demand video streaming. LiveShift is an application that combines both approaches. While video is transmitted through the peer-to-peer network in a live fashion, all peers participate in a distributed storage. This adds ability to replay time-shifted streams from other peers in a distributed and scalable manner. For the demonstration, a decentralized network is used, with peers running on EMANICSLab nodes and notebook computers.

1 Introduction

The popularity of devices with IP connectivity, combined with growing reachability of such networks, increases the demand for video transmission over the Internet [5]. Since video streaming requires large bandwidth, peer-to-peer (P2P) technology has been successfully applied to reduce bandwidth cost and add scalability by [4, 3] and others.

There are two categories in which current implementations fall [4]. (1) On a Video-on-Demand (VoD) streaming, a user may choose the content and play it at any time. Current applications include YouTube and the P2P-based Joost. (2) With live streaming, a user may choose among different channels. Current popular applications are SOPCast and Zattoo, both based on P2P.

Time-shift [6] is a feature that allows a live stream viewer to watch the stream in a particular point in the past. However, it is a challenge as it requires large storage space and a peer alone can only record the channel it is currently watching.

LiveShift is a time-shift-enabled P2P live streaming application. A user watching a live transmission has the abil-

ity to pause the transmission and continue at a later moment. Moreover, a user is able to watch recorded content from other peers; thus, enabling time-shifting for many live streams in a distributed and scalable fashion. Peers in the P2P network represent individual users watching a particular video stream. Each peer is responsible for recording and storing parts of the video stream. Time-shift in LiveShift consists on finding peers responsible for the recording of the requested time slots and retrieving the video stream. A P2P live streaming application with distributed time-shift does not exist to our knowledge.

2 LiveShift

LiveShift is a software prototype designed to take full advantage of P2P networks for both transmission of live video and storage of past video data. Figure 1 shows its architecture.

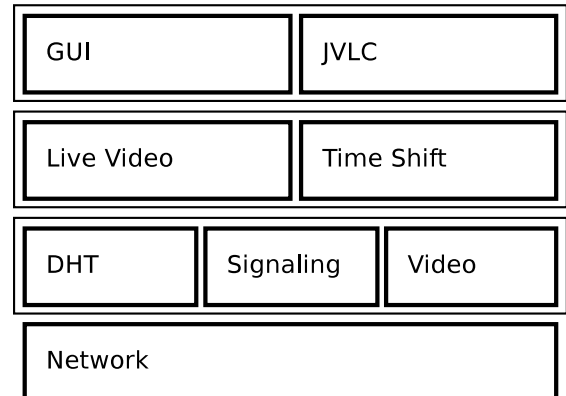


Figure 1. LiveShift Basic Architecture

The DHT is used to maintain the list of available channels and sources, with information such as schedule and bitrate. Any peer can be a broadcaster, creating a new channel. As

peers join the channel and store segments, the information is updated. The DHT also contains information about video data stored on peers for time-shifting. Signaling and video data are transferred using UDP.

In order to receive the live stream, a multiple tree layered approach is used, similar to [3]. The video data is split by the source into different layers, and each layer is propagated through a different tree. This approach aims to minimize negative effect on receivers during churn.

To store the video data for time-shifting, video packets must be modified by the source peer to include timing information. Each layer is divided into segments of fixed duration. Peers receiving a live or time-shifted stream store the video in a memory buffer. When the buffer reaches a predefined size of one segment, the segment then may be stored on a long-lasting storage. After storing the segment, the peer adds a reference to the segment in the DHT. The video is only recorded by peers that are currently watching the stream, incurring no extra overhead.

Figure 2 shows the LiveShift GUI. Users are able to choose a channel to watch live content. Pushing the pause button freezes the image but keeps recording the stream on the local peer. These recordings can be watched by pressing the play button. The stop button interrupts the reception of the stream. With the drop down boxes, a user can specify the timeshift parameters for streams not recorded locally. Additionally, information about the overlay is shown in a separate window, in order to observe the P2P overlay. VLC Media Player [1] is used for encoding, decoding, and playing video.



Figure 2. The GUI of LiveShift

3 Demonstration Scenario

The demonstration scenario is composed of 16 peers. Three peers stream each a video channel, while 13 other peers receive and watch these channels. Each receiving peer watches one channel at a time. Channel switching is possible. 14 peers are deployed in EMANICSLab, which is based on MyPLC [2], while 2 peers are deployed on notebook computers, showing a streaming peer and a receiving peer.

LiveShift is presented as follows. First, video is captured during the presentation of the demo and can be watched live through the P2P network. Then, during the demonstration, the time-shift function is used to replay parts of the presentation and video that has been captured beforehand. The time-shift can be disabled, which will result in watching again the live stream.

LiveShift is implemented in Java version 1.6 and runs on any platform supported by the Java Virtual Machine and VLC [1], for example, Linux, MacOS and Windows 98/NT/XP/Vista. VLC allows peers to exchange control messages over direct socket connections with other peers. VLC encodes video streams and sends it over the network. Once a network connection is established, UDP is used to transmit video packets, which is then sent through the local loopback interface, to the media player, which decodes the stream.

References

- [1] VLC media player. <http://www.videolan.org/vlc/>, 2008.
- [2] MyPLC. <http://www.planet-lab.org/doc/myplc>, 2008.
- [3] Y. Cui and K. Nahrstedt. Layered peer-to-peer streaming. In *NOSSDAV '03: Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video*, pages 162–171, New York, NY, USA, 2003.
- [4] T. Do, K. Hua, and M. Tantaoui. P2vod: providing fault tolerant video-on-demand streaming in peer-to-peer environment. *ICC 04: IEEE International Conference on Communications*, 3:1467–1472, June 2004.
- [5] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross. A measurement study of a large-scale p2p iptv system. *IEEE Transactions on Multimedia*, 9(8):1672–1687, December 2007.
- [6] T. Wauters, W. V. de Meerssche, F. D. Turck, B. Dhoedt, P. Demeester, T. V. Caenegem, and E. Six. Co-operative proxy caching algorithms for time-shifted iptv services. *SEAA 06: 32nd Euromicro Conference on Software Engineering and Advanced Applications*, pages 379–386, August 2006.