Chapter 11
A Self-Organized Structured Overlay Network for Video Streaming

Khaled Ragab
King Faisal University, Saudi Arabia

ABSTRACT

Peer-to-Peer (P2P) file downloading and streaming applications have recently attracted a large number of users on the Internet. Currently, several P2P video streaming systems have been deployed to reduce the cost at server. They are classified into two categories live and on-demand streaming systems. The live streaming systems disseminate live video contents to all peers in real time. On the other hand, the on-demand video (VoD) streaming system enables peers to enjoy the flexibility of watching video. It realizes the goal of watch whatever you want, whenever you want. The current P2P-VoD systems cannot realize such goals efficiently. This chapter proposes a self-organized structured overlay network of peers to realize that goal and improve performance. Each peer is able to cache some video minutes associated with the current media being played. The proposed overlay network is organized into clusters. Each cluster contains peers with overlapped buffer windows where their playing points are located between lower and upper play point limits. When a peer in the cluster moves its play point within the limits, for example by performing a seek operation, it then can rapidly discover and fetch the required blocks for the playback buffer from peers in the same cluster. Clusters improve both discovery and fetch overheads. However, it needs cluster management overhead.

INTRODUCTION

The video streaming applications have recently fascinated a large number of users over the Internet. According to AccuStream iMedia Research, the number of video streams served increased 52% in 2008, and reaches 34 billion viewers. In addition, over the last ten years audiences have accessed 142.7 billions of pieces of video (Accustream 2009). According to comScore, in July around 91 million viewers watched five billion online video on
*YouTube*. Americans watched 558 million hours of online during that month (comScore 2009).

The most familiar solution for streaming video over the Internet is the client-server service model such as *YouTube*. This model uses variants of technologies such as Content Delivery Network (CDN) to push video content from server to its clients through delivery servers. These servers are placed strategically at the network edges. A CDN replicates content from the origin server to cache servers, scattered over the globe, in order to deliver content to end-users in a reliable and timely manner from nearby optimal surrogates (Mukaddim Pathan and Rajkumar Buyya 2008). Scalability is the major challenge for server based video streaming solutions as follows. Despite the fact that *Google* has its impressive CDN, on 23 November 2008, it relied on a third party Akamai, to stream a *YouTube* live concert to 700,000 concurrent viewers. Akamai technologies evolved out of an MIT research effort aimed at solving the flash crowd problem. Even with today’s low bandwidth Internet video of 400 kbps, the *YouTube* live concert needed more than 280Gbps server and network bandwidth. Akamai, the largest commercial CDN service provider reports a peak aggregate capacity of 200Gbps with its tens of thousands of servers (Akamai 2007). Actually, the *Google* and Akamai CDN infrastructures are not necessary to handle live video streaming effectively. Instead Peer-to-peer (P2P) video streaming systems can handle it effectively and far cheaper because peers are serving most of the video to others. P2P file downloading and streaming applications have recently attracted a large number of users on the Internet. The main difference between P2P file sharing and P2P streaming applications is the instant time when content is used. In P2P file sharing, content is completely transferred before files are opened. In contrast to P2P streaming where content is decoded, played immediately and later (probably) is discarded. Currently, several P2P video streaming systems have been deployed to reduce server cost. They are classified into two categories: live and on-demand streaming systems. The live streaming systems disseminate live video contents to all peers in real time. On the other hand, the on-demand video (VoD) streaming system enables peers to enjoy the flexibility of watching. It realizes the goal of watch whatever you want, whenever you want. The current P2P-VoD systems cannot realize such goals and provide poor performance. For example, if a peer requests a video server to download one hour MPEG1 video (i.e. 540MB) using modem 28.8Kbps, 10Mbps then it has to wait 42 hours, 7.2 minutes to watch the video.

Enormous television networks, news sites, and video sharing sites such as *YouTube* and Google Video provide VoD. Most of the VoD being delivered today is short-length, low bit-rate clips. For example, *YouTube* videos today are typically less than 10 minutes in length and have a bit rate under 200 kbps. In the near future, we expect a high demand for higher bit rate (potentially DVD quality) and longer videos (including full length movies) streamed over the Internet. P2P-VoD is a new challenge for the P2P technology. Moreover, P2P-VoD has less synchrony in the peers sharing video content compared to streaming live video content. Consequently it is more difficult to reduce load at server while maintaining the streaming performance. Thus, to reduce load at video server this chapter presumes that each peer contributes a small storage for replicating video content. In addition, the video server divides the video data into segments. Each segment maintains video data of five minutes video.

This chapter reviews the state-of-the-art of peer-to-peer technologies for video streaming, and presents taxonomy of various solutions that have been developed. Moreover, this chapter proposes a careful and efficient design to replicate and discover video content. It proposes a self-organized structured overlay network of peers that organizes peers into clusters. Each cluster contains peers with close playing points and overlapped buffer windows. Each peer identifies its neighbors based