Centralized Peer-to-Peer Streaming with PFGS Video Codec

Ivan Lee[1] and Ling Guan[2]

[1] School of Electrical and Information Engineering, University of Sydney, NSW, Australia
ivanlee@ieee.org

[2] Department of Electrical and Computer Engineering, Ryerson University, ON, Canada
lguan@ee.ryerson.ca

Abstract. This paper addresses the bottleneck of the conventional video streaming using the client-server approach, and an innovative centralized peer-to-peer approach is proposed. In order to achieve the best perceived quality of service, combination of layered source coding and distributed network adaptation is proposed in this paper. The proposed technique features centralized management, guaranteed perceived quality of service, while offloading the bottleneck traffic loads among the peers. The performance of the centralized peer-to-peer streaming protocol is benchmarked against the differential service technique, by using the H.264 video codec with the progressive fine granular scalable extension.

1 Introduction

With the growing popularity of the personal computer industry, highly computerized world leads to the demand of computer interactions. Popular applications such as video conferencing and video on demand require effective video transmission over the internet. Video streaming, which requires higher network bandwidth and possesses tighter real-time constraints, is one of the enabler technologies for such applications.

Today, most of the video streaming techniques are based on the client/server architecture, where a centralized server is responsible for the entire transmission requests. Typically, this dedicated streaming server possesses longer uptime, higher bandwidth, and higher processing power. However, the client/server architecture does not scale to the growing audiences. In addition, the current internet is a heterogeneous collection of networks, and the QoS guarantee is missing by default. These characteristics present the bottleneck for video streaming using the client/server approach. To address these limitation, numerous techniques in transmission feedback control, adaptive source encoding algorithm, efficient packetization, resource allocation, and error control coding have been proposed to improve the quality of the video communication on today's Internet [1] [2]. Popular approaches such as traffic prioritization using DiffServ [3] and transmission bandwidth reservation using RSVP [4] are widely studied and realized in some commercial products.

K. Aizawa, Y. Nakamura, and S. Satoh (Eds.): PCM 2004, LNCS 3331, pp. 131–138, 2004.

Distributed media streaming is another approach to address the limitations of client/server based streaming. A popular information exchange approach in the P2P framework is the simultaneous streaming from multiple senders. This approach yields higher throughput and increases the tolerance to loss and delay due to congestion [5]. The P2P multimedia streaming and caching service also reduces the initial delay for the playback, and minimizes the delay jitter during playback [6]. Deshpande et al. [7] proposed an application-layer multicast architecture called SpreadIt. The requesting nodes are formed peering layers on-demand, which builds up a multicast tree. Each node within the tree is responsible for forwarding the data to its descendants. Like typical multicast applications, SpreadIt does not restrict to video payload data, and it is designed for real-time delivery of any data type. SpreadIt aims to reduce the loss during the transmission, and there is no quality guarantee when video is transmitted using its framework. Tran et al. [8] also investigated application-layer multicast tree building, and proposed the ZIGZAG algorithm featuring short end-to-end delay, low control overhead, efficient join and failure recovery, and low maintenance overhand.

Centralized Peer-to-Peer Streaming Protocol (P2PSP) was first introduced to combine layered multimedia codec with distributed streaming infrastructure [9]. The performance of P2PSP compares to the client-server approach was evaluated using a 3D-DCT based video codec with vector quantization [10]. In this paper, we analyze the performance of centralized Peer-to-Peer Streaming Protocol (P2PSP), benchmarked against the DiffServ technique. H.264 FPGS is the video codec under our investigation.

2 Client/Server Versus Centralized Peer-to-Peer Streaming

The Client/Server network is the de-facto architecture for video streaming, because historically clients do not possess sufficient bandwidth or computational power to forward requests from neighbourhood peers. Client/Server streaming is illustrated in Fig. 1(a), and the protocol is summarized in Fig. 2(a). The Client/Server architecture consists of several limitations:

1. **Single point of failure** - the centralized streaming server is the only mean for data distribution, and it is therefore the target for the attack to deteriorate the service. While duplicating the servers and providing redundant network infrastructure could help to resist from the failures, however, it is not cost effective and is vulnerable to a large scale Denial of Service (DoS) or Distributed DoS (DDoS) attacks.
2. **Scalability problem** - despite the fact that techniques such as multicast was proposed to utilize the network resources, it is not designed for non-synchronized media streaming. This problem escalates when interactive media streaming applications are widely available to the audiences.
3. **Inefficient utilization of the network resources** - the Client/Server architecture was derived from historical computing infrastructure where huge

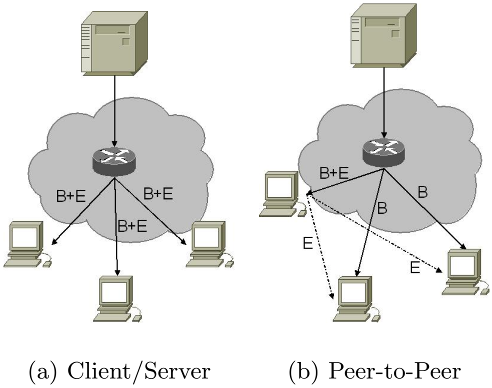


Fig. 1. The network topology for the simulation, where B denotes the base-layer stream and E denotes the enhancement layer streams

and expensive mainframe computers serves thin clients or dumb terminals. With the rapid growth of the computing and networking industries, the clients are capable of high computational task and the broadband access to the Internet has become affordable, and such trend will continue in the foreseeable future. Deploying these computational power and bandwidth within the Client/Server framework will underutilize the available resources.

The Peer-to-Peer (P2P) network empowers the peers to act the server's function to communicate between each other. In a fully decentralized P2P network, theneed of a central server is completely eliminated. Video streaming over the P2P network presents a feasible alternative to the Client/Server architecture, for offloading the traffic from the bottleneck link to the underutilized links. Although the P2P architecture has the potential to overcome some limitations raised by the Client/Server architecture, the P2P system possesses different constraints due to the following characteristics:

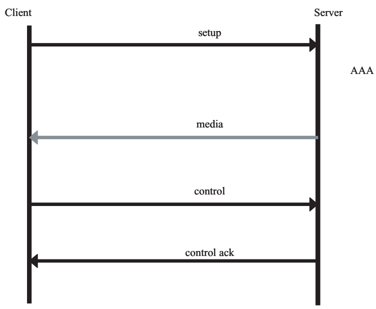
– **Heterogeneous**: Different peers have different bandwidth and processing capacity.
– **Unpredictable**: Each peer is free to join and leave the service at any time. There is no reinforcement to guarantee the service from the peers.

Both the heterogeneous and unpredictable behavior can severely impact the quality of the streaming video. Therefore, we propose a hybrid client/server and P2P streaming infrastructure, named centralized P2PSP. Centralized P2PSP uses the neighbourhood peers to offload the traffic load from the bottleneck link, while the server's roles in centralized management for authentication, authorization, and accounting are reserved. Centralized P2PSP is design for streaming any media type coded in base/enhancement layers, as illustrated in Fig. 1(b). The detailed protocol handshake for centralized P2PSP is shown in Fig. 2(b).

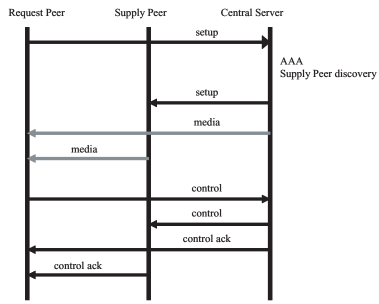
Step 1. The request peer P_{req} sends a setup message to the central server. The server firstly validates the AAA for P_{req}. Once the P_{req} is successfully validated, the server forwards the request to a nearby supplying peer P_{sup}. If no nearby peer has cached the video content, the server will also act as P_{sup}.

Step 2. P_{req} send the request signal for the video. The server commences streaming base-layer video, whereas P_{sup} is responsible for streaming the enhancement-layer video. The server updates P_{req} in the database.

Step 3. The server and P_{sup} continuously streams the video contents to P_{req}, until the media control signal (such as pause, play) or the session termination signal is received from the P_{req}.



(a) Client/Server network



(b) Peer-to-Peer network

Fig. 2. Handshake of the video streaming for Client/Server and peer-to-peer network

3 P2PSP with PFGS

The H.264 PFGS coding consists of a base-layer and multiple enhancement layers. The enhancement layers are based on a low-complexity DCT-like bit-plane scheme. Different quality levels are achieved by decoding different levels of layers. In PFGS, each base-layer is always predicted by the base-layer from the previous frame, whereas an enhancement layer is predicted from the second or third enhancement layer of the previous frame. Details of the PFGS algorithm can be found in [11]. High coding efficiency and flexible scalability makes PFGS ideal evaluating centralized P2PSP.

As discussed previously, the client/server streaming model consists a network bottleneck. Let n_r denotes the number of receivers, r_{base} denotes the average transmission rate for the base layer video, and r_{enh} denotes the enhance layer video. The average bandwidth requirement of the bottleneck link b_{cs} is illustrated in Equation (1).

$$b_{cs} = n_r(r_{base} + \sum_{\forall EnhLayers} r_{enh}) \tag{1}$$

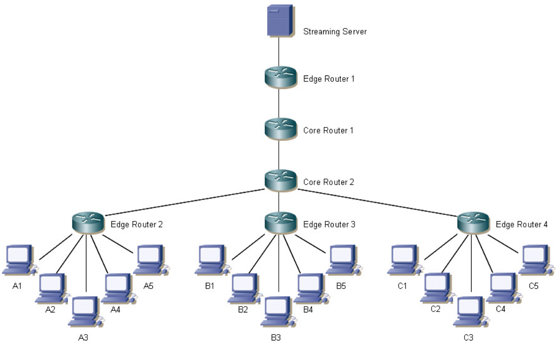


Fig. 3. Network topology for the simulation

For P2PSP, since the streamer server does not supply enhancement layers, the average bandwidth requirement at the bottleneck link, b_{p2psp}, is shown in Equation (2).

$$b_{p2psp} = n_r r_{base} + \sum_{\forall EnhLayers} r_{enh} \tag{2}$$

The traffic reduction on the bottleneck link is therefore:

$$b_{cs} - b_{p2psp} = (n_r - 1) \sum_{\forall EnhLayers} r_{enh} \tag{3}$$

Equation (3) indicates that the average bandwidth requirement for client/server and P2PSP approaches will only be equal (1) if there is only one receiving node (n_r equals 1), or (2) there is no enhancement layers ($\sum r_{enh}$ equals 0). For any other scenario, video transmission with the P2PSP approach will outperform the client/server approach.

A common technique for improving the streaming quality is to buffer the traffic. Let t denotes the time, t_0 denotes a reference starting time, and t_b is the buffering period. Let $r_{base,t}$ denotes the base-layer transmition rate at time t, and $r_{enh,t}$ denotes the enhancement-layer transmission rate at time t. The average bandwidth requirement $b_{cs,t}$ for client/server model is:

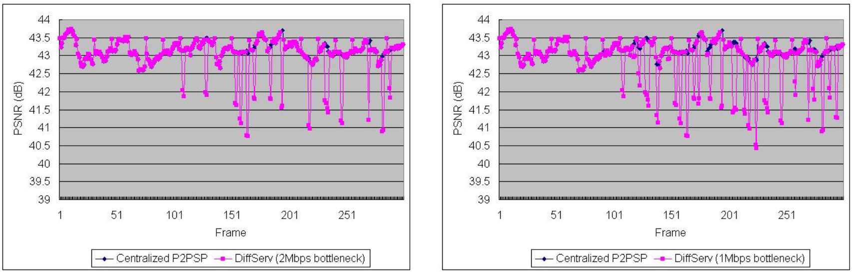
$$b_{cs,t} = \frac{n_r}{t_b} \sum_{t=t_0}^{t_0+t_b} (r_{base,t} + \sum_{\forall EnhLayers} r_{enh,t}) \tag{4}$$

Similarly, the average bandwidth requirement $b_{p2psp,t}$ for P2PSP model is:

$$b_{p2psp,t} = \frac{n_r}{t_b} \sum_{t=t_0}^{t_0+t_b} r_{base,t} + \frac{1}{t_b} \sum_{t=t_0}^{t_0+t_b} r_{base,t} \sum_{\forall EnhLayers} r_{enh,t} \tag{5}$$

The average bandwidth saving on the bottleneck link by using P2PSP with buffering is:

$$\frac{n_r - 1}{t_b} \sum_{t=t_0}^{t_0+t_b} \sum_{\forall EnhLayers} r_{enh,t} \tag{6}$$



(a) Bottleneck bandwidth 2Mbps (b) Bottleneck bandwidth 1Mbps

Fig. 4. Video quality with different bottleneck bandwidth

(a) DiffServ video output: frame 138 (PSNR=41.335dB) (b) Centralized P2PSP video output: frame 138 (PSNR=42.759dB)

Fig. 5. Output frame comparison for DiffServ and P2PSP

4 Simulation

The simulation is performed using NS-2, with the network topology shown in Fig. 3. The bottleneck link is located between Core Router 1 and Core Router 2, with the buffering period t_b in Equation. 4 and 5 equals 5ms. In the client/server mode, the streaming server transmits the entire video traffic to each of the receiving nodes. DiffServ is experimented in the client/server mode. Each video frame is packetized in a fixed size packet, to generate CBR traffic for the simulation.

Fig. 4 shows the comparison of the receiving video quality between P2PSP and DiffServ, under different bottleneck bandwidth. Both Fig. 4(a) and Fig. 4(b) indicate that the buffering technique at the bottleneck link reduces the packet dropout rate, evidenced by no packet dropout initally before the buffer is fully loaded. Once the buffer is over-loaded, DiffServ start to dropout packets and therefore degrades the received video quality. Comparing Fig. 4(a) and Fig. 4(b), we observe that when the bottleneck bandwidth is reduced, DiffServ experiences more severe quality degrade. In comparison, P2PSP always result equal or better video quality. An example frame output is shown in Fig. 5.

5 Conclusions

In this paper, we proposed a distributed approach for video streaming. Centralized Peer-to-Peer Streaming Protocol addresses the bottleneck of the conventional Client and Server Streaming technique, to achieve higher perceived quality of service. Compares to competing technologies such as DiffServ, centralized P2PSP does not require software or hardware upgrade on the existing internet backbone. Centralized P2PSP offloads video traffic from the bottleneck link and decentralizes the processing power from the streaming server, while maintaining the high integrity in terms of authentication, authorization, and accounting. Simulations results show that centralized P2PSP can provide a consistent video quality at the receiver site, while DiffServ starts to dropout packets and degrades the video quality. We conclude that implementing centralized P2PSP video streaming is a cost-effective and highly scalable solution.

References

1. Dapeng Wu, Yiwei Thomas Hou, Wenwu Zhu, Hung-Ju Lee, Tihao Chiang, Ya-Qin Zhang, and H. Jonathan Chao, "On end-to-end architecture for transporting mpeg-4 video over the internet," *IEEE Trans. Circuits Syst. for Video Technology*, 2000.
2. Dapeng Wu, Yiwei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang, and Jon M. Peha, "Streaming video over the internet: Approaches and directions," *IEEE Trans. on Circuits and Systems for Video Technology*, 2001.
3. S. Blake, D. Black, M. Carlson, E. Davies, Z.Wang, and W.Weiss, "An architecture for differentiated services," in *IETF RFC 2475*, December 1998.
4. Lixia Zhang, Steve Deering, Deborah Estrin, Scott Shenker, and Daniel Zappala, "Rsvp: A new resource reservation protocol," in *IEEE Network Magazine*, September 1993.
5. T. Nguyen and A. Zakhor, "Distributed video streaming over the internet," in *Proc. of SPIE Conference on Multimedia Computing and Networking*, Jan 2002.
6. Won J. Joen and Klara Nahrstedt, "Peer-to-peer multimedia streaming and caching service," in *Proc. of IEEE International Conference on Multimedia and Expo*, Aug 2002.
7. H. Deshpande, M. Bawa, and H. Garcia-Molina, "Streaming live media over a peer-to-peer network," Tech. Rep. 2002-21, Stanford University, March 2002.
8. Duc A. Tran, Kien A. Hua, and Tai T. Do, "Zigzag: An efficient peer-to-peer scheme for media streaming," Tech. Rep., University of Central Florida, 2002.
9. Ivan Lee and Ling Guan, "A scalable video codec design for streaming over distributed peer-to-peer network," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, November 2002.

10. Ivan Lee and Ling Guan, "Centralized peer-to-peer streaming with layered video," in *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, July 2003.
11. Yuwen He, Feng Wu, Shipeng Li, Yuzhuo Zhong, and Shiqiang Yang, "H26l-based fine granularity scalable video coding," in *Proc. ISCAS*, May 2002.