*2011 International Conference on Software and Computer Applications*
*IPCSIT vol.9 (2011) © (2011) IACSIT Press, Singapore*

# An Interactive Streaming Service over Peer-to-Peer Networks

Yu-Wei Chen [1] and Yu-Hao Huang [2] +

[1] National Taipei University of Technology, Graduate Institute of Information and Logistics Management, Taipei, Taiwan

[2] National Taipei University of Technology, Industrial Engineering and Management, Taipei, Taiwan

**Abstract.** It is a significant challenge to provide user interactivity in peer-to-peer on-demand video streaming, such as pause, fast search and reverse search operations of VCR function. In order to provide this type of interactive services, system need to serve asynchronous user requests of large number, adjust the P2P overlay topology immediately and allocate the streaming appropriately. In this paper, we propose a Tree-based Interactive MultimEdia System (TIMES). Each peer in TIMES maintains a neighbor list and hybrid caching strategy. Consequently, TIMES can significantly reduce the demand of server bandwidth, increase system flexibility and provide the interactive services.

**Keywords:** peer-to-peer network, interactive multimedia system, video-on-demand, caching

## 1. Introduction

The so-called Video-On-Demand (VoD) service allows users to select and watch programs of their interest. Today, the transmission of multimedia and entertainment has become an indispensable part of people’s life. However, providing VoD service still has two challenges [1]: First, bandwidth is an expensive resource for transmission. However, VoD is provided by the transmission of multimedia. It’s requires higher bandwidth requirements and the latency will be higher. Second, when a VoD server can provide more and more popular content, the more users will attempt to join that system. In the traditional client-server architecture, each client connects to the server over a dedicated channel. When the video popularity increases, system performance will decrease.

Multicast [2][3] is a well-known technique to reduce server loading and network bandwidth. In typical multicast transmission, when video server transmits a multicast stream, the clients in the multicast group can receive the stream simultaneously. Compared with unicast, multicast can significantly reduce the server loading.

In order to effectively reduce the load on the server, P2P technology has been proved to be an effective and flexible method for video streaming in dynamic and heterogeneous environments [4][5][6]. In the past few years, P2P based methods have had great success in application over the internet, such as file sharing, file download and streaming transmission. Successful examples include PPLive [5] and Coolstreaming [7]. In addition, some research has been aimed at non- synchronization requests and has proposed some “cache-and-relay” methods [8], such as client caches where it has recently played data and then uses the cached content to serve others. Some researchers have even used this concept for the P2P streaming system [9][10][11]. So recently, many systems combine P2P technology and VoD services to reduce the cost of video transmission effectively [12]. Unfortunately, supporting interactive operations such as pause, fast forward and rewind functions [5][6] are still at a bottleneck in development.

---

+ Corresponding author. Tel.: + (886-2)27712171#2364; fax: +(886-2)2731-7168.
*E-mail address*: ywchen@ntut.edu.tw (Yu-Wei Chen), puzpazboom@hotmail.com (Yu-Hao Huang)

In this paper, we propose a Tree-based Interactive MultimEdia System (TIMES). By maintaining hybrid caching strategy, clients can not only watch an entire video without interruption but also use interactive operations such as pause, fast jump and rewind jump.

The rest of this paper is organized as follows: we introduce related works in Section 2. In Section 3, we present the TIMES. In Section 4, we present the conclusion of this paper.

## 2. Related Works

In this section, we divide the research of VoD services into the following three categories, as detailed below:

### 2.1. Typical client-server VoD services

This kind of transmission is when a client wants to watch some video content, that client will send a request to the video server. The server then will establish a dedicated channel for the client and send the complete video stream to each client directly when it receives a request from any client. In HON [13] a hybrid approach combined a tree overlay and gossip-based data exchange method. However, all the relationships between peers and gossip partners are assigned by a central server in HON. Therefore, the central server needs to keep track of all the users. So this client-server architecture is not preferred in P2P systems because a lot of control traffic would congest the bandwidth of any central server.

### 2.2. VoD combined with P2P technology

Unlike the previous types of architecture, this type of research will join the concept of P2P into VOD service and hope clients can play a role between client and server simultaneously, so all clients can share the bandwidth loading with the server. PROP [14] uses self-organization and a distributed storage approach to provide VoD service. But PROP relies on proxy servers and global information for catching replacement. P2VoD [15] divides overlay structure into layers based on the peer's joining time, each peer will store the recent video date, so P2VoD can support asynchronous requests. In P2Cast [1], clients are clustered according to their arrival times and build an application multicast tree. The server has to patch the missing part for each late arriving client. The proposal found in [16] proposes the concept of a chunk group to reduce the loading on the server. The peers in groups are using P2P technology and sharing the video data with each other.

### 2.3. VoD combined with P2P technology and interactive VCR functions

In recent years, most concepts of interactive streaming service were proposed and hoped to provide some interactive functions like pause, fast forward or rewind and other interactive features. RINDY [17] designed a ring topology structure to support VCR function under VoD structure. Peers in RINDY can use neighbor lists to find a suitable position during the random access process. DTStream [18] used a derivative tree-based overlay management scheme to support peer interactivity in streaming system. VMesh [19] is a distributed P2P VoD streaming scheme. In VMesh, videos are divided into segments and stored with each peer to improve the segment supply. So it can support efficient interactive commands.

## 3. Tree-based Interactive MultimEdia System

In this section, we first show some observations on VoD user behavior. Then we'll introduce our system structure in detail and present a hybrid caching strategy

### 3.1. Observations on VoD user behavior

The existing work also has an implicit assumption that the process of watching has continuity of playback from begin to end. However, the proposals [22][23] had the following observations, thus we base our methods. First, most videos are visited partially, so such as fast forward, rewind of VCR functions are used quite frequently. Second, from the statistics, the jump distances in watching process are usually small. Third, some parts are often accessed, such as the score screen in a football game are always attracting most audience to watch it.

## 3.2. System Overview

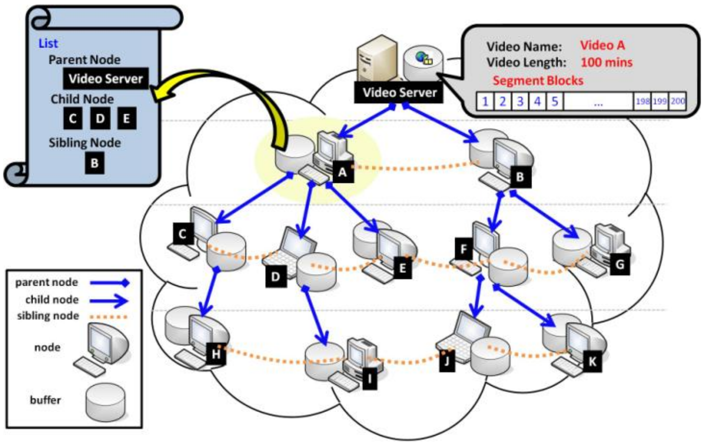
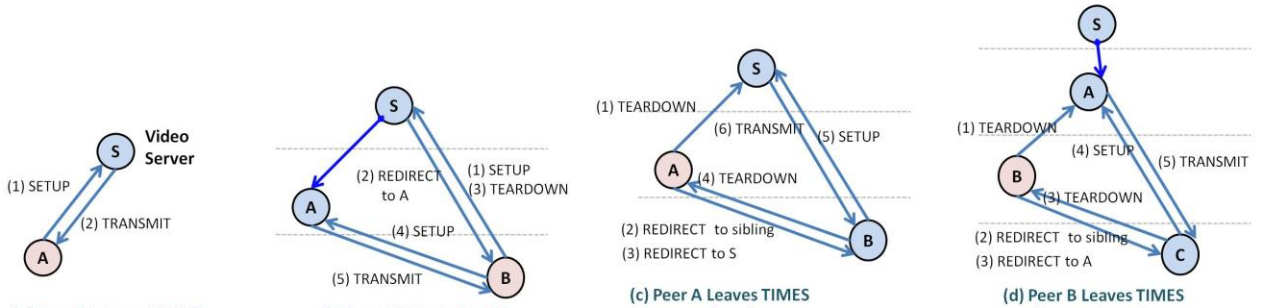


Fig. 1: System Overview

Figure 1 illustrates the architecture of TIMES, we only consider single video transmission. The initial process in the system, video A will divide into many segments, each length is equal to *k*. According to a peer's joining time, we separate these peers into different layers, and the interval between each layer is equal to *k*. Earlier joining nodes will belong to the upper layer and subsequent nodes will send their request to earlier joining nodes rather than connect with server directly. Besides, each node will store a neighbor list to record adjacent node information, including the parent node, sibling node and child node. Based on this list, peers can find the target peer who has its requested data much more easily. In TIMES, every parent node uses multicast streaming to transmit the video segment to its child nodes. So if the request rate is much higher, the server need only send one multicast streaming to peers of the first layer, and these peers will cache the segment and relay to their child nodes in next layer. Thus, large number of non- synchronization requests can be satisfied.

## 3.3. Tree Architecture

In this part we will further explain the general peer operations like 'join,' 'leave' and the operations of VCR such as pause, fast-jump and rewind-jump.

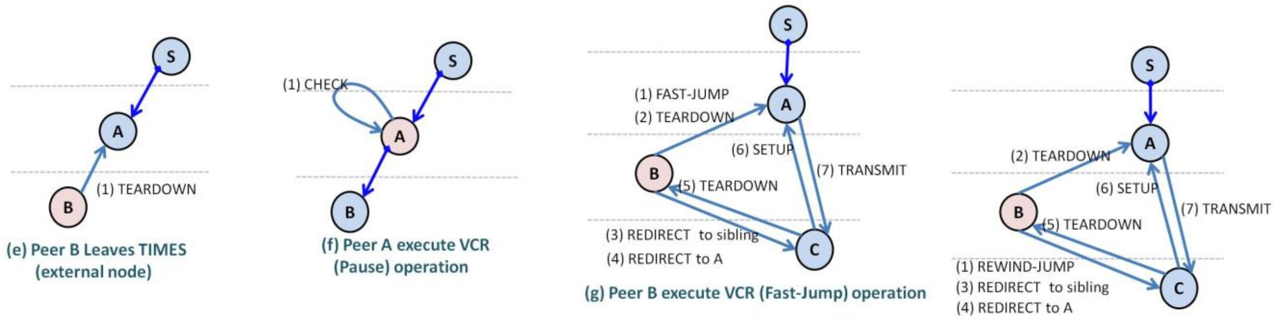


(a) Peer A joins to TIMES

(b) Peer B joins to TIMES

(c) Peer A Leaves TIMES (root)

(d) Peer B Leaves TIMES (internal node)



(e) Peer B Leaves TIMES (external node)

(f) Peer A execute VCR (Pause) operation

(g) Peer B execute VCR (Fast-Jump) operation

(h) Peer B execute VCR (Reverse-Jump) operation

Fig. 2: Peer's Operations in TIMES

Figure 2 shows the peer's operation for every case. If there's no peer in the system, peer A will connect to the server directly in figure 2(a). In figure 2(b), because peer A is already in this system, so if peer B sends a connect request to the server, it will be rejected and the response redirects the message to peer B to let it redirect the request message to peer A on its own. If a peer wants to leave the VoD system, we consider three cases for peers leaving. In figure 2(c), if a root wants to leave the TIMES, peer A should send the redirect message to its child node first, then if peer A has another sibling node, its child node should connect with peer A's sibling node, else peer A's child node will directly connect to the server. In figure 2(d), if the leaving peer B is an internal node, the leaving process is similar to the former case, but the second choice for peer C is to connect with peer B's parent node rather than the server. In figure 2(e), if the leaving node is an external node as peer B, it can leave directly after it sends the teardown message to its parent node.

In figure 2(f), if peer A wants to execute a pause operation, peer A will still receive segments from the server during the pause process and check its buffer space continuously. If peer A's buffer is not enough to store the data, peer A should execute a rejoin process. In figure 2(g), if peer B wants to fast-jump to a specific position, peer B should send the request to ask its upper-layer relatives to find a new provider via neighbor list exchanges, Rewind-jump is similar with fast-jump, peer B sends the rewind-jump message to its child node or lower-layer relatives as per figure 2(h).

### 3.4. Hybrid Caching Strategy

In TIMES, each node, in so much as possible, uses its own limited buffer space to catch video segments and relay the data to its child node, so the system can transfer between peers and peers to satisfy the various peer demands. However, in the actual environment, each node can provide only limited buffer space, so if now it has a peer that wants to join this system, but tall the peers in this system replace the former part of segments because their buffer space is not enough. Finally, this peer will find no suitable parent peer and redirect its request to the server. So if a buffer's replacement strategy is poorly designed, it will greatly reduce the system's performance. Thus we propose a hybrid caching strategy and divide peer's buffer space into two parts, one is the initial part and another is the current part.

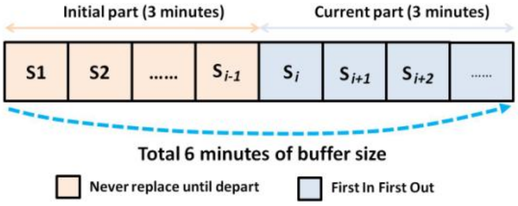


Fig. 3: Hybrid Caching Strategy

Figure 3 illustrates the hybrid caching strategy in TIMES. In the initial part, we design it so that each peer caches the first initial 3 minutes of video and never replaces this part until it depart form TIMES. We consider that most of the subsequent peers usually miss the initial part, so we let each peer keep storing the initial part of the video to avoid the initial missing of data. On the other hand, according to the former concept what we proposed, each subsequent peer usually uses an earlier entry peer's buffer space to receive the target segments smoothly even if it executes some VCR operation. So we design it so that each peer caches the current 3 minutes of the video and then uses the First-In-First-Out algorithm to replace the buffer space when it's full.

## 4. Conclusion and Future Work

In this paper, we proposed a Tree-based Interactive MultimEdia System (TIMES). Each peer in TIMES maintains a neighbor list and hybrid caching strategy. Consequently, TIMES can significantly reduce the demand of server bandwidth, increase system flexibility and provide the interactive services. In future work, we will design a mesh-based interactive VoD system to improve the system stability.

## 5. References

[1] Y. Guo, K. Suh, J. Kurose, and D. Towsley, "P2Cast: peer-to-peer patching scheme for VoD service," in Proceedings of the 12th ACM International World Wide Web Conference (WWW), Budapest, Hungary, May 2003.

[2] K.C. Almeroth and M.H. Ammar, "The use of multicast delivery to provide a scalable and interactive video-on-demand service," *IEEE J. Selected Areas in Communication*, vol. 14, no. 6, pp. 1110-1112, 1996.

[3] U. Shaukat, A. Qayyum and S. Hussain, “Performance evaluation of multicast protocols for multimedia traffic,” in *Proceedings of the 2009 Int'l Conference on Emerging Technologies*, October 2009.

[4] “PPLive”, http://www.pplive.com/.

[5] Y. Huang, T. Z. J. Fu, D. M. Chiu, J. C. S. Lui and C. Huang, “Challenges, design and analysis of a large-scale P2P-VoD system,” in *Proc. of ACM SIGCOMM’08*, August 2008.

[6] C. Huang, J. Li, and K. W. Ross, “Can internet video-on-demand be profitable?,” in *Proc. ACM SIGCOMM*, Kyoto, Japan, August 2007.

[7] X. Zhang, J. Liu, B. Li, and T.S.P. Yum, “Donet/Coolstreaming: A datadriven overlay network for live media streaming,” in *Proc. IEEE INFOCOM*, 2005.

[8] S. Jin and A. Bestavros, “A cache-and-relay streaming media delivery for asynchronous clients,” in *Proc. Int. Workshop on Networked Group Communication (NGC)*, Boston, Massachusetts, USA, October 2002.

[9] Y. Cui, B. Li, and K. Nahrstedt, “oStream: Asynchronous streaming multicast in application-layer overlay networks,” *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 91–106, January 2004.

[10] A. Sharma, A. Bestavros, and I. Matta, “dPAM: A distributed prefetching protocol for scalable asynchronous multicast in P2P systems,” in *Proc. IEEE INFOCOM*, Miami, USA, March 2005.

[11] Y. Shan and S. Kalyanaraman, “Hybrid video downloading/streaming over peer-to-peer networks,” in *Proc. IEEE International Conf. on Multimedia & Expo (ICME)*, Baltimore, Maryland, July 2003.

[12] S. Annapureddy, S. Guha, C. Gkantsidis, D. Gunawardena, and P. R. Rodriguez, “Exploring VoD in P2P swarming systems,” *IEEE INFOCOM 2007*, 2007.

[13] M. Zhou and J. Liu, “A hybrid overlay network for video-on-demand,” in *Proceedings of IEEE International Conference on Communications (ICC)*, Seoul, Korea, May 2005.

[14] L. Guo, S. Chen, and X. Zhang, “Design and evaluation of a scalable and reliable P2P assisted proxy for on-demand streaming media delivery,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 5, pp. 669–682, May 2006.

[15] T. T. Do, K. A. Hua, and M. A. Tantaoui, “P2VoD: Providing fault tolerant video-on-demand streaming in peer-to-peer environment,” in *Proc. of IEEE ICC’04*, June 2004.

[16] R. Takano and Y. Yoshizawa, “Offloading VoD server organized dynamically distributed cache using P2P delivery,” *International Conference on Information Networking*, pp. 1-5, January 2008.

[17] B. Cheng, H. Jin, X. Liao, “Supporting VCR functions in P2P VoD services using ring-assisted overlays,” in *Proc. of IEEE ICC’07*, June 2007.

[18] T. Xu, J. Chen, W. Li, S. Lu, Y. Guo, and M. Hamdi, “Supporting VCR-like operations in derivative tree-based P2P streaming systems,” in *Proc. of IEEE ICC’09*, Dresden, Germany, June 2009.

[19] W. P. K. Yiu, X. Jin, and S. H. G. Chan, “VMesh: Distributed segment storage for peer-to-peer interactive video streaming,” in *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1717-1731, December 2007.

[20] C. Xu, G. M. Muntean, E. Fallon, and A. Hanley, “A balanced tree-based strategy for unstructured media distribution in P2P networks,” in *Proc. of IEEE ICC’08*, June 2008.

[21] Y. He and Y. Liu, “VOVO: VCR-oriented video-on-demand in large-scale peer-to-peer networks,” in *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 20, no. 4, pp. 528-539, 2009.

[22] H. Yu, D. Zheng, B. Zhao, and W. Zheng, “Understanding user behavior in large-scale video-on-demand systems,” *ACM SIGOPS Operating Systems Review*, vol. 40, no. 4, pp. 333-344, 2006.

[23] C. Zheng, G. Shen, and S. Li, “Distributed pre-fetching scheme for random seek support in peer-to-peer streaming applications,” in *Proc. ACM Workshop on Advances in Peer-to-Peer Multimedia Streaming*, Hilton, Singapore, November 2005.