A Framework for Cloud P2P VoD System based on User's Behavior Analysis

A. M. Riad
Information Technology Dept.
Faculty of Computers and Information, Mansoura University, Egypt.

Mohammed Elmogy
Information System Dept.
Faculty of Computers and Information, Mansoura University, Egypt.

A. I. Shehab
Information System Dept.
Faculty of Computers and Information, Mansoura University, Egypt.

ABSTRACT

Video distribution over the Internet has already become a major application due to users' growing demand of Video content and continuous growth of network technologies. Recently, as the rapid growth in the number of Video-On-Demand (VoD) systems, people ask for more and more video resources, which cause overload on network resources. Therefore, how to use the capability of peers to reduce the server's load and how to maintain the streaming performance for Peer-to-Peer (P2P) technology. One major quality of video delivery metric is playout latency which can be considered as one of the key challenges that face many researchers nowadays. On the other hand, as cloud computing grows rapidly and more Internet users prefer to watch video or listen to audio online instead of downloading them to their local disk, it is critical and important to provide a cloud data storage and media delivering services continuously to deal with VoD systems.

To address the above issues, this paper proposes a novel cloud framework for P2P VoD system based on the analysis of the user's behavior. The proposed framework based on double buffering technology with hybrid caching strategy which is built upon a cloud architecture. Simulation results of the proposed framework are presented and discussed.

Keywords:

Video-On-Demand (VoD), peer-to-peer (P2P) networks, buffering, multimedia communications, Disk cache.

1. INTRODUCTION

Video-over-IP can be considered as one of the top Internet applications which has recently attracted a large number of users over the Internet. In 2013, Over 6 billion hours of video are watched each month on YouTube. Youtube [1] alone hosted more than 45 terabytes of videos, More than 1 billion unique users visit YouTube each month, and 100 hours of video are uploaded to YouTube every minute.

Therefore, Video-On-Dema3nd (VoD) has become one of the most popular Internet application. Many years ago, traditional VoD systems used the client-server architecture and delivered all streaming data by centralized media servers, in which the bandwidth of these servers often turned to be the bottleneck for the whole system. In contrast, Peer-to-Peer (P2P) video streaming protocols, motivated by the great success of file sharing applications, have attracted a lot of interests from academia and industry. A major advantage in using P2P is that each peer involved in a content delivery contributes with its own resources to other peers in the session [2].

On the other hand, several new challenging problems have come along with the widespread applications of P2P technology in VoD. Most P2P applications are neither adaptable nor friendly to Internet Service Provider (ISP). P2P approach is more scalable and needs less investment as it makes full use of clients' resource (bandwidth, storage, CPU, etc). Therefore, each client is not only a service consumer, but also a service provider. Then, system management and quality of service (QoS) under peer churn are high challenges. Peers usually perform selfishly. They only care with their own benefits and ignore the requirements of the other peers [3].

There are many factors that cause missing data requests and hence the peer will have to download the missing chunks from the server. These factors can be summarized as [4]: (1) insufficient upload capability, (2) firewall or proxy, (3) new media content, or (4) peer departure. On the other hand, the keys to the popularity of the Internet-based video are immediate, high quality, distortion-free, continuous playout. The primary metrics for measuring user experience are [5]: the initial playout latency (i.e., the amount of time between pressing the play button and seeing video rendered on the screen) and the video artifacts (i.e., distortion, pixelation, or interruption in the rendered video). Accordingly, minimizing waiting time is dependent upon the network's ability to deliver video data. Higher throughput allows faster initialization of the video buffer, while a minimum throughput causes playout interruptions.

Otherwise, buffering a video stream allows you to watch online videos as playing local video from your local drive. The keys of successful buffering lie in the speed of the Internet connection, the amount of RAM being used by the player computer, the amount of available memory on the computer, and the efficiency of the video player [3].

Nowadays, as the fast growth in population of p2p VoD system, users' behavior is playing an increasingly crucial role in the performance of video system. Therefore, user watching and sharing behavior become much more important for system performance than ever before. The server load is affected by the user's watching behavior. Also, the sharing efficiency of p2p network is determined by user store and sharing behavior [6].

On the other hand, the cloud computing paradigm has recently been receiving lots of attention since it emerged. It represents an important step towards the dream that all aspects of computation can be organized as a public utility service [7, 8]. Cloud computing has the advantage of powerful service capacity, high availability, and high scalability.

In this paper, a P2P VoD scheme based on the analysis of the user's behavior is presented. A double buffering technique with data storage mechanism is presented. It would upload valuable chunks to super nodes in the network for increasing their availability for future sharing. This mechanism would increase the availability of chunks and also it will overcome the playout latency problem. Therefore, it would minimize the waiting time for watching a requested video. The main contribution of this paper is how to pre-distribute the video contents before requesting it. The proposed framework will be discussed in detail in section 3.

The rest of this paper is organized as follows: Section 2 presents an overview of the current P2P streaming systems. It also presents a depth survey on the existing techniques of video streaming over P2P. In section 3, the proposed framework will be discussed. The simulation results of the proposed double buffering scheme on different modes will be elucidated in section 4. Finally, section 5 concludes our proposed framework and offers some thoughts on the future of the video delivery research.

2. RELATED WORK

Last years, the P2P based video streaming have attracted the interests of both researchers and academia. There are some solutions that have been proposed to address different challenging issues in P2P technology in VoD. In this section, some of these well known solutions will be presented. At the beginning, the P2P based VoD systems can be classified into three main types: named tree - based [9], mesh - based [10], and hybrid systems. Tree-based P2P VoD systems are popular and used in many implementations, such as, o Stream [11]. on the other hand, mesh- based P2P overlays have been designed in many scenarios such as pcVoD [12], PROMISE [13], and many other systems.

For example, TAG [14] presented a hybrid overlay, which was based on tree-based gossip protocol. It uses a balanced binary tree to index all joined peers and find many partners from the tree when a peer joins.

Directory based P2P VoD architecture is proposed to support VCR operations [15]. The directory server works as a central administrator to maintain a database that keeps track of all servers and clients, and helps new clients to locate the required service. Meanwhile, the content servers store contents and serve clients' requests. Although the scheduling packet is small, this is a centralized solution which needs a strong administrative server to handle global scheduling and do not provide inter- channel overlay optimizations. In all these P2P-based VoD systems, several media data caching schemes are proposed. But, all of these systems used only one buffer that caches the current watching chunks and there is no multiple channels support in their design [12, 15].

Cui and Nahrstedt [16] addressed the data distribution issue in large-scale VoD overlay networks by using the strong buffering capacity of end hosts, i.e., peers. An interesting point in their proposal is that they declared to defeat the server's theoretical lower bound of traditional multicast-based schemes. Their system was robust against unexpected requests. They defined several complicated operations but there was no disk caching support in their design.

Some other researches combine tree structure and mesh structure to schedule media data in VoD environment, such as TAG [14]. . This work increased the deployment difficulty and is still suffering from discontinuous playback, as a high-level peer's departure is significantly influencing the successor peers.

A peer cached VoD service (pcVoD) is proposed with the aid of a tracker server by Ying [12]. This is similar to some P2P file-sharing applications. The media data routes in the way improved from file-sharing. Thus, such systems require multiple copies of a resource, in order to ensure a demanding peer with enough downloading speed from several providing peers. This assumption will fail if the resource is not popular or VCR operations frequently occur.

Guo et al. [17] used self-organization and a distributed storage approach to provide VoD service. Their system is called PROP which relies on proxy servers and global information for catching replacement. P2VoD [18] 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. P2Cast [18], 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 [19] 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.

Recently the concept of cloud computing has been widely spread, some researchers turn to carry out studies on the resource management issues above the cloud. There is an interest in the research community in issues arising from running computation-intensive and data-intensive applications on clouds [20].

Park [21] considers the issue of introducing a brokering system where agents bid for resources, which is supported on an adjustable quality video encoding scheme. Su [22] presents a multi-user streaming system that adjusts the encoding rate of each stream. Foster [23] describes the GARA resource management architecture which aims to achieve end-to-end QoS guarantees across heterogeneous collections of shared resources.

There is also related research that has put insight into adaptive QoS models in wireless networks or distributed networks. Since the concept of cloud computing has been widely spread, some researchers tend to carry out studies on the resource management issues above the cloud [24].

3. THE PROPOSED FRAMEWORK

In traditional P2P streaming systems, memory-based buffering scheme is normal one and just for single channel overlay. In order to transmit media data on the Internet, the media data should be divided into clips. Usually there are two typical methods: one is to divide the media file into clips which have the same playing time, the other one is to divide the media file into clips which have the same size [25]. But using the second method can easily cause the problem that the different media streams cannot synchronize into one stream to play. To avoid the problem, the media files are cut into some basic units, which are usually 1 sec. each unit called a chunk or a slot, which is the basic unit of data for peers.

To best of our knowledge, in current systems, a peer only caches chunks currently being watched in only one channel. This limits the use of hard disk and memory resources; also deleting files after they are played eliminates many later opportunities for future sharing. Our framework proposes a double buffering scheme which is dependent on user's behavior gathered by the tracker. It depends on both buffer and disk caching when delivering chunks. The overall architecture of the proposed system is shown in Figure 1. The detailed streaming mechanism is presented in the sequent subsections.

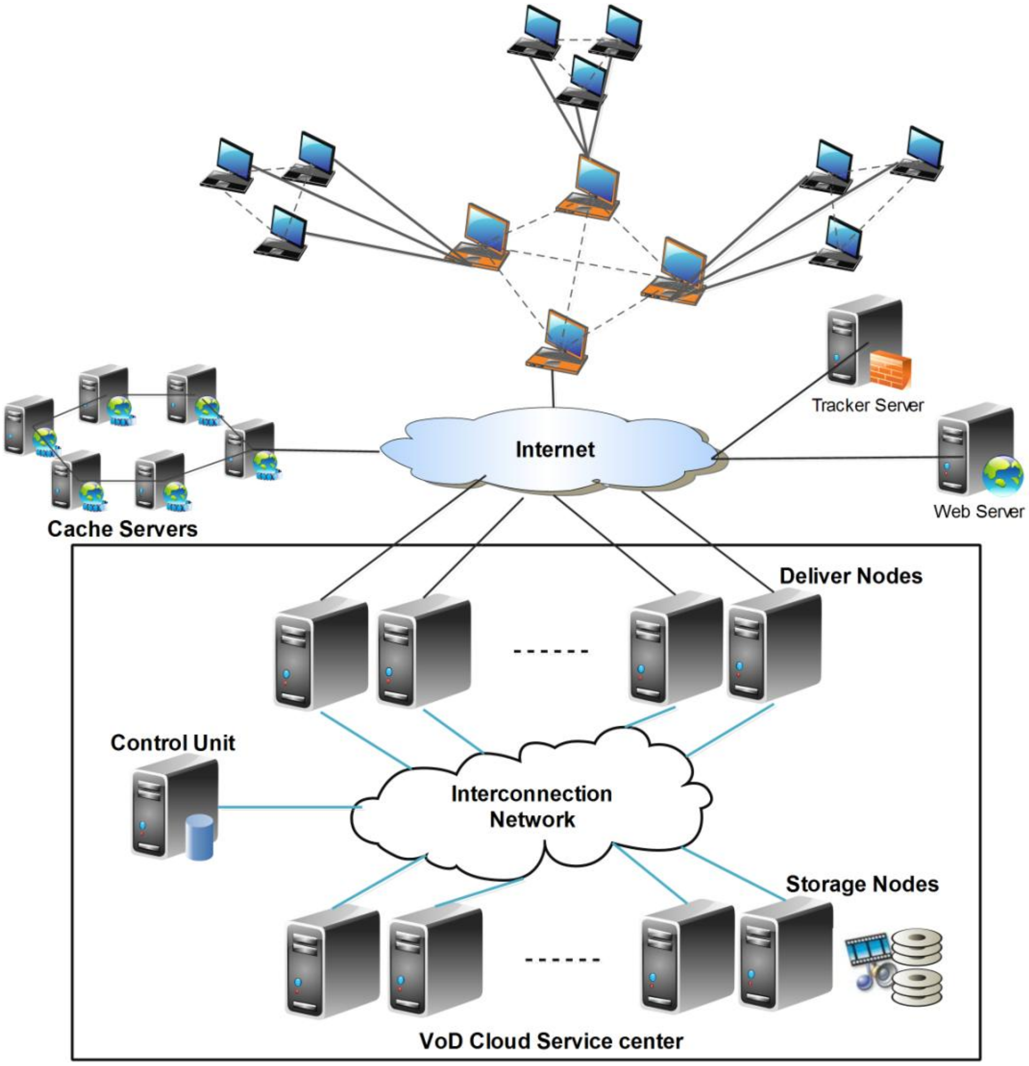

Fig 1: The architecture of the proposed cloud P2P VoD system.

3.1 VoD Cloud Service Center

As shown in Figure 1, the rectangle depicts a VoD cloud service center comprised of:

1) Storage Nodes: A storage node is used to provide large-scale store capacity in order to meet the users' requirements of video and audio contents which are clustered and located on the disks of storage nodes.

2) Deliver Nodes: A deliver node is used to deliver streaming media data to peers directly. If currently a deliver node doesn't find the required media content in its memory, it will start to send a request to corresponding storage node.

3) Control Unit: The control unit responsible for monitoring and scheduling. It periodically collects the running status of all nodes, and keeps the information for later use. Also it receives peers' requests and tries to schedule them to a certain deliver node. Besides, it also keeps all the information about media location on the disks of storage nodes. The control unit should direct a deliver node wants to obtain data from a certain media file absent in its memory to a certain storage node.

3.2 P2P Overlay

Outside the cloud service center, as shown in Figure 1, the proposed system includes a web server, a tracker server, a caching layer server and peers organized in a tree-mesh overlay.

1) Web server, it provides the latest updated channel list to incoming peers. The user goes to the portal, browses the catalog of videos, and then selects one to watch.

2) Tracker server, it is responsible for indexing the basic information of all online peers, and behaves as a well-known rendezvous to provide initial members a list for each joining peer, which facilitates data sharing between peers. The tracker maintains a membership list of all joined peers. This information is used to facilitate data sharing between peers and need not be perfect for the system to function correctly. It provides the latest updated channel list to incoming peers. The user goes to the tracker, browses the catalog of videos, and then picks one to watch.

3) Caching layer, In order to guarantee the QoS of end-users and speeds access for required media, we add one or more caching server(s) which has large memory capacity to cache the hot-spot media contents for users. These servers are used only if there is no space neither on peers' disks nor on delivering nodes.

4) Peers, download chunks from cloud or neighbor peers, and then cache these chunks on local disk. In the proposed system, Multiple Video Caching is supported so that each peer could not only share data of current channels but also cached data of previously popular channels. The peer has two components, the peer client and the media player. They communicate using the real-time streaming protocol (RTSP) over a local socket.

The participating peers in the overlay network are organized into two layers: one layer contains super peers and the second contains ordinary peers. Tracker classifies peers into super or ordinary peers according to their uplink bandwidth. In fact each peer should follow a set of steps when requesting a specific content that are:

(1) Peer contacts the web server to browse the catalog and select a video file.

(2) Web server returns the selected video file's ID to the peer, redirecting it to the tracker.

(3) P2P tracker redirecting it either to Super peer, ordinary peer, cloud, or caching server.

(4) Chunks are fetched to the destination.

3.3 Double Buffering Mechanism

In general, on-demand services enable clients to watch videos from beginning to end. As long as clients are able to buffer the initial part of the video, they are watching. On-demand service can provide access to the video to the next clients that request to watch that video. Therefore, the key challenge is how to keep the initial part of a video in a peer's buffer for the longest possible period. which will maximize the availability of a video for stream relay.

The framework proposes a multi-channel overlay depending on users behavior gathered by the tracker. Each peer has only one main channel that represents the currently watched video and the peer could be attained into several service channels. For example, Figure 2 shows that peer P6's main channel is channel 1, and has two service channels: channel 2 and channel 3. Peer P4's (or P5) main channel is channel 1 and service channel is channel 3. P9's main channel is channel 2, and has no service channel.

If the user is usually watching continually, another videos after the end the current video. Here, the tracker applies a heuristic algorithm that seeks for the high ranked favorite videos and then the system opens two buffers on the peer side: one sending the current watching chunks and the other sending the initial parts of recommended videos.

On the other hand, selecting the high ranked videos (high access count, related videos, and history of other peers) is based on current peer's behavior analysis through monitoring the recent most visited videos and also through taking into account other peers' experiments in the overlay.

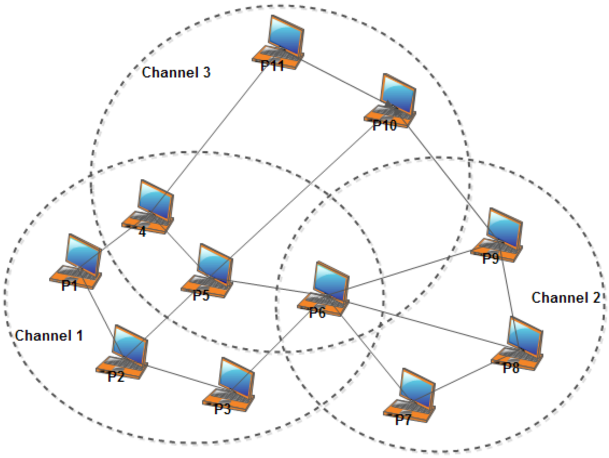

Fig 2: The proposed multi-channel overlay.

The tracker is also responsible for suggesting videos that are correlated with the current watching video cluster. Figure 3 demonstrates double buffering scheme, the destination peer has two buffers: (1) watching buffer and (2) service buffer.

As shown in Figure 3, the *watching buffer* receives video packets from sender peers P1 and P2 where the service buffer receives three high ranked videos from peers P1, P3, and P4. It is noticed that a sender peer (P1) have chunks from two different videos and hence can send chunks for both buffers at the destination peer. The chunks that are not currently used are not removed but instead moved to the storage media disk for future use. If the user select one of the already cached videos in the service buffer, chunks would immediately passed to the decoder and in turn displayed.

By this way, peer will watch the video with zero waiting time (no latency). Furthermore, in worst cases, if the peer does not select any video of those already cached in the service buffer, service buffer remains servicing other peers in the overlay need to watch any of these videos. In such mechanism a destination side scheduler with the aid of tracker server is implemented to manage the delivery of different parts of the video from different sender peers.

The double buffering mechanism aims to reduce the server load without increasing user waiting time (i.e. minimizing playout latency). When requested content has already been pre-distributed and exists in the service buffer, the user can watch the requested content without requesting it from the video server and without generating traffic on the network. Therefore, by pre-distributing high ranked video content, we can greatly reduce the load of the video server and the network.

In order to take advantage of disk space, when the media clips are eliminated from the buffer, the data are cached onto the disk. The caching space is provided by peers and total space will be listed into a K -bytes index of disk slots. Index list S is created to manage the states of the disk slots.

When the media packet with length L is eliminated from the memory, the destination peer has to find out N disk slots, which are continuous and free, to store the media packets L. A t that time, the following condition should be met [25]: $N \geq 1$ and $K \times N \geq L$.

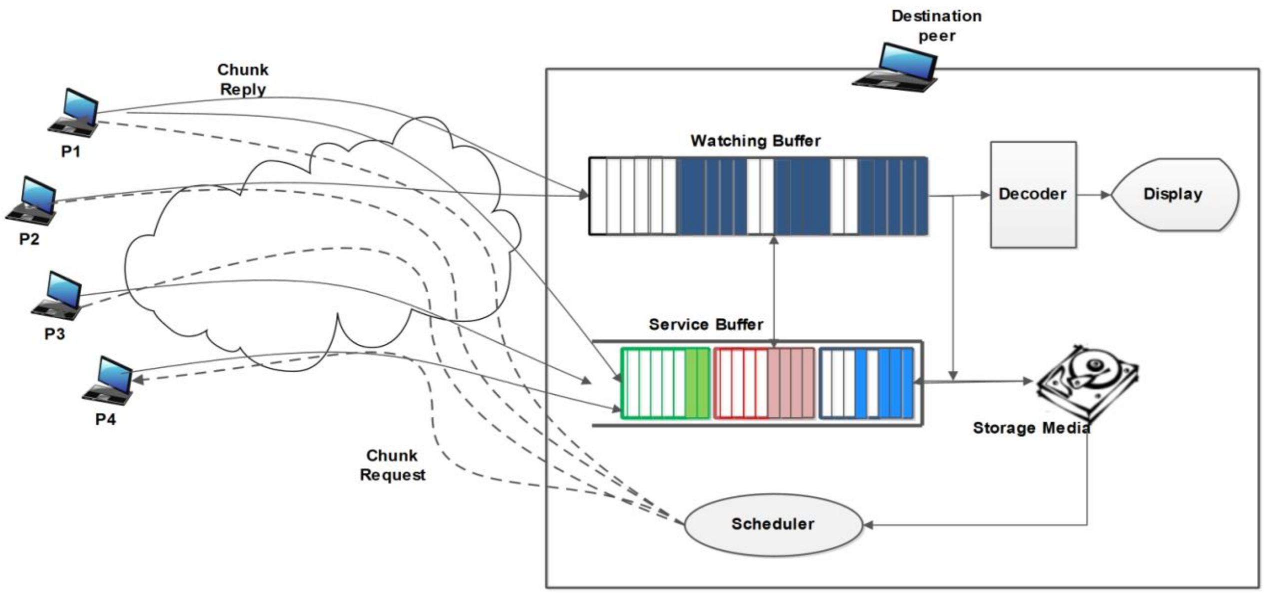

Fig 3: The proposed double buffering mechanism.

If the N disk slots are found, information set S (D, L , P) will be inserted into the end of a written buffer where D means the disk slot position , L means the length of the disk slot and P means the beginning number of the disk slot found. Otherwise, it means that the disk caching space is too full to cache more data and the old data on the disk should be replaced with a new data.

4. SIMULATION RESULTS AND DISCUSSION

In the following subsections, the performance of the proposed double buffering scheme is evaluated by testing the model in different modes.

The OPNET Modeler 17.1 simulation tool was selected as the tool of choice given its widespread adoption in both commercial and military domains. OPNET is a research oriented network simulation tool. It is a very powerful software tool that simulates the real world behavior of wired and wireless networks.

4.1 Simulation Parameters

4.1.1 Network Topology

To make the study more realistic and representative of a real world, we try to simulate the proposed model in Mansoura university. The proposed model consists of two geographically separated subnets: VoD subnet and peers' subnet. The VoD subnet is located in communication and information technology center and it provisions a VoD server capable of streaming stored video content to participating peers on request. The peers are located in faculty of computer sciences. The arrival of peer requests follows the Poisson distribution with a mean of λ, where λ ranges from 0 to 100 (requests/min.). Additionally, a peer's forwarding capacity is set to be equal to its downloading capacity. In other words, a peer is capable of forwarding as many descriptions as it receives.

Both subnets are connected to the Internet cloud via DS3 WAN circuits. The approximate distance between the two subnets is 10 km.

The VOD subnet resides on a 100Mbps IP network that connected to an access router which is connected to the Internet via a 45 Mbps DS3 WAN link. The ADSL configuration employed in this model has a 3.0 Mbps downlink channel and a 640 kbps uplink channel. Moreover, the Internet "cloud" was configured with a packet discard ratio of 0.001% which results in one packet out of every 100,000 packets is dropped in the Internet.

4.1.2 Video Traffic

The first traffic source is an actual video trace from a 10 minute MPEG-2 movie clip of Terminator 2 which utilized a 1280x720 frame format resolution and a 30 fps encoding rate. This traffic was obtained from Arizona State University [26] and it has a high resolution format and encoding frame rate.

The second traffic source was a 2-hour MPEG-4 Matrix III movie trace [26] which utilized a 352x288 frame format resolution and a 25 fps encoding rate.

4.2 Performance Evaluation

In the evaluation process, the network bandwidth of each peer is set to 20–100 MB, he memory size for caching data is 300 sec, and disk cache size is 200 MB.

In an attempt to measure the server load in both single buffer and the proposed scheme (|Double Buffering), the model was simulated using the second traffic source. Figure 4 shows that the server load using double buffering mechanism is less than using a single buffer.

The packet end-to-end delay measured in the simulation run on one peer is detailed in Figure 5.

The two curves are averaged across the 2 hour movie. From the results, the reader can notice

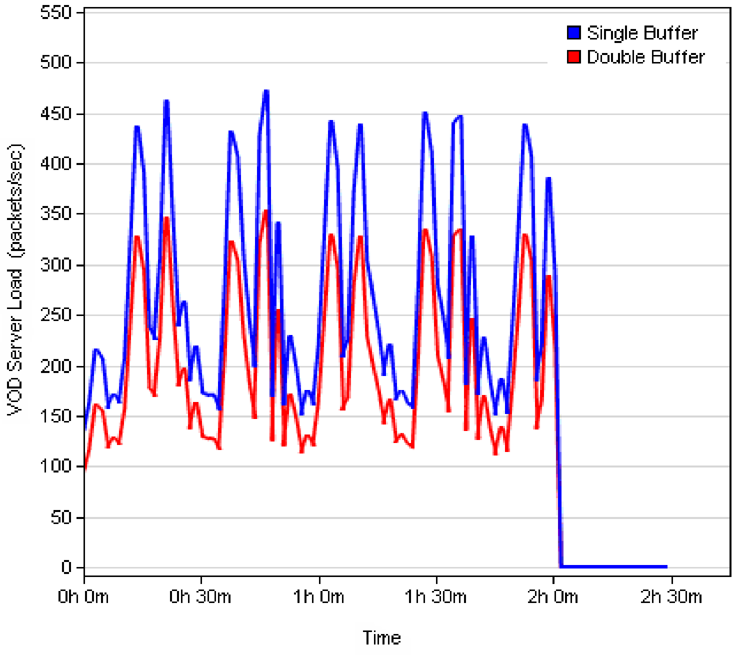

Fig 4: VOD Server Load

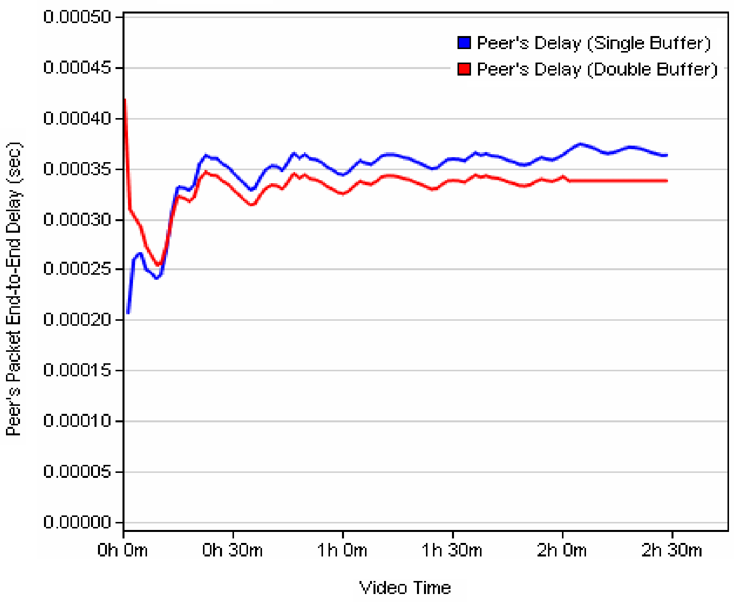

Fig 5: Packet End-to-End Delay

that the two curves are closely tracked each other about 35 ms, In general, double buffering scheme achieve less delay compared to a single buffer scheme.

5. CONCLUSION AND FUTURE WORKS

Based on the analysis of current existing P2P VoD systems, this paper presented a strategy and an implementation of double buffering scheme to buffer media data in a P2P VoD system. Our main contribution is based on users' behaviors analysis in order to pre-fetching high ranked videos. Taking storage media into account besides buffering also help reducing traffic on the network, minimizing server load, and speeds up the process of fetching chunks into buffer. The experiments show that the server load and packet end-to-end delay are significantly reduced. Further research work will be on how to detect the malicious peers in p2p system. Some participating peers may not cooperate as desired. They may be selfish and unwilling to upload data to others, or they may have abnormal actions. All these uncooperative, abnormal or attacking behavior are called malicious actions and the associated peers are malicious peers. Also, further research work will be on how to extract hot chunks in a video and sending these chunks instead of initial part of a video.

6. REFERENCES

[1] Liu, Y.; Guo, Y. & Liang, C. (2008), 'A survey on peer-to-peer video streaming systems', *Peer-to-Peer Networking and Applications* **1**(1), 18-28.

[2] Ramzan, N.; Park, H. & Izquierdo, E. (2012), 'Video streaming over P2P networks: Challenges and opportunities', *Image Commun.* **27**(5), 401--411.

[3] Lu, Z.; Wu, J.; Chen, L.; Huang, S. & Huang, Y. (2010), CPH-VoD: A Novel CDN -P2P-Hybrid Architecture Based VoD Scheme, *in* Lei Chen; Peter Triantafillou & Torsten Suel, ed., 'Web Information Systems Engineering â€“ WISE 2010', Springer Berlin Heidelberg, , pp. 578-586.

[4] Liao, X.; Jin, H. & Yu, L. (2012), 'A Novel Data Replication Mechanism in P2P VoD System', *Future Gener. Comput. Syst.* **28**(6), 930--939.

[5] Ma, K. J.; Bartos, R. & Bhatia, S. (2011), 'Review: A survey of schemes for Internet-based video delivery', *J. Netw. Comput. Appl.* **34**(5), 1572--1586.

[6] Zheng, Y.; Peng, J.; Yu, Q.; Huang, D.; Chen, Y. & Chen, C. (2010), A measurement study on user behavior of p2p VoD system, *in* 'Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on', pp. 373-376.

[7] Wang, X.; Du, Z.; Liu, X.; Xie, H. & Jia, X. (2010), An adaptive QoS management framework for VoD cloud service centers, *in* 'Computer Application and System Modeling (ICCASM), 2010 International Conference on', pp. V1-527-V1-532.

[8] M. Armbrust, A. Fox, R. G. e. a. (2009), 'Above the clouds: A Berkeley view of cloud computing', *EECS Department, University of California, Berkeley. Tech. Rep.*.

[9] Liao, X.; Jin, H.; Liu, Y. & Ni, L. (2007), 'Scalable Live Streaming Service Based on Interoverlay Optimization', *Parallel and Distributed Systems, IEEE Transactions on* **18**(12), 1663-1674.

[10] Zegura, E.; Calvert, K. & Bhattacharjee, S. (1996), How to model an internetwork, *in* 'INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE', pp. 594-602 vol.2.

[11] Cui, Y.; Li, B. & Nahrstedt, K. (2004), 'oStream: asynchronous streaming multicast in application-layer overlay networks', *Selected Areas in Communications, IEEE Journal on* **22**(1), 91-106.

[12] Ying LH, B. (2005), A pcVOD: internet peer-to-peer video-ondemandwith storage caching on peers, *in* Canada, ed., 'In Proceedings of the Eleventh International Conference on Distributed MultimediaSystems'.

[13] Hefeeda, M.; Habib, A.; Botev, B.; Xu, D. & Bhargava, B. (2003), PROMISE: peer-to-peer media streaming using CollectCast, *in* 'Proceedings of the eleventh ACM international conference on Multimedia', ACM, New York, NY, USA, pp. 45--54.

[14] Liu, J. & Zhou, M. (2006), 'Tree-assisted gossiping for overlay video distribution', *Multimedia Tools Appl.* **29**(3), 211--232.

[15] Guo Y, Suh K, K. J. T. D. (2003), A peer-to-peer ondemand streaming service and its performance evaluation., *in* 'In Proceedings of ICME'.

[16] Cui, Y.; Li, B. & Nahrstedt, K. (2006), 'oStream: asynchronous streaming multicast in application-layer overlay networks', *IEEE J.Sel. A. Commun.* **22**(1), 91--106.

[17] Guo, L.; Chen, S. & Zhang, X. (2006), 'Design and evaluation of a scalable and reliable P2P assisted proxy for on-demand streaming media delivery', *Knowledge and Data Engineering, IEEE Transactions on* **18**(5), 669-682.

[18] Do, T.; Hua, K. & Tantaoui, M. (2004), P2VoD: providing fault tolerant video-on-demand streaming in peer-to-peer environment, *in* 'Communications, 2004 IEEE International Conference on', pp. 1467-1472 Vol.3.

[19] Takano, R. & Yoshizawa, Y. (2008), Offloading VoD Server Organized Dynamically Distributed Cache Using P2P Delivery, *in* 'Information Networking, 2008. ICOIN 2008. International Conference on', pp. 1-5.

[20] Pandey, S.; Wu, L.; Guru, S. & Buyya, R. (2010), A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments, *in* 'Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on', pp. 400-407.

[21] Park, H. & van der Schaar, M. (2007), Congestion game modeling for brokerage based multimedia resource management, *in* 'Packet Video 2007', pp. 18-25.

[22] Su, G.-M. & Wu, M. (2005), 'Efficient bandwidth resource allocation for low-delay multiuser video streaming', *IEEE Transactions on Circuits and Systems for Video Technology,* **15**(9), 1124-1137.

[23] Foster, I.; Kesselman, C.; Lee, C.; Lindell, B.; Nahrstedt, K. & Roy, A. (1999), A distributed resource management architecture that supports advance reservations and co-allocation, *in* 'Quality of Service, 1999. IWQoS '99. 1999 Seventh International Workshop on', pp. 27-36.

[24] Stillwell, M.; Schanzenbach, D.; Vivien, F. & Casanova, H. (2009), Resource Allocation Using Virtual Clusters, *in* 'Cluster Computing and the Grid, 2009. CCGRID '09. 9th IEEE/ACM International Symposium on', pp. 260-267.

[25] Yu, L.; Liao, X.; Jin, H. & Jiang, W. (2011), 'Integrated buffering schemes for P2P VoD services', *Peer-to-Peer Networking and Applications* **4**(1), 63-74.

[26] Auwera, G. V. D.; David, P. T. & Reisslein, M. (), 'Traffic characteristics of H.264/AVC variable bit rate video,? Submitted to', *IEEE Communications Magazine* 2007.