On-line Dynamic Buffer Allocation For VoD

M Dakshayini* **A S Manjunath****

*** Research Scholar, Dr.MGR University, faculty Dept. of ISE,BMSCE,**
Member, Multimedia Research Group, Research Centre, DSI, Bangalore.
**** CEO & MD, Manvish eTech Pvt, Bangalore.**

***Abstract*-Due to high buffer and bandwidth requirement and rate variability of compressed video, delivering video across the wide area networks is a challenging issue. Proxy servers have been used to reduce bandwidth requirement and improve client access time on the internet by caching the passing data. In this paper, we propose a unique VoD architecture and On-line dynamic buffer allocation algorithm for efficient buffer and bandwidth management. This scheme reduces the bandwidth requirement between the main multimedia sever and the Proxy servers. Also reduces the client waiting time by sharing the videos among the proxy servers of the Local proxy servers group, by allocating and then reallocating the buffer at Proxy sever (PS) dynamically based on the current frame size. The proposed scheme results in efficient usage of buffer and bandwidth for the videos, very low request rejection ratio. And also results in low client waiting time by balancing the load among the neighboring proxy servers and non neighboring proxy servers of LPSG. And reduction of load on the main multimedia server by storing more number of frequently used videos at PS. Our simulation results shows very high buffer and bandwidth utilization, acceptable waiting time for user requests in all conditions, load sharing among the proxy servers and between the main multimedia server and proxy server with tracker's coordination.

Key words: VoD architecture**,** main multimedia server [MMS], Proxy server [PS], Local proxy server group [LPSG], buffer, client waiting time.

1. Introduction

VoD is possible because of tremendous advancement in internet and multimedia technology. Applications like movie on demand, distant learning, news on demand etc are some of the applications of VoD. The high buffer and bandwidth requirement, and rate variation of videos in compressed format introduces some challenging issues to end-to-end delivery over wide area network(WAN). WAN is usually shared by many institutions or organizations across distant regions and thus its buffer and bandwidth is more expensive in WAN, than in LAN. A VoD system usually has several servers and distributed clients over the entire network. These servers contain prerecorded videos and are streamed to the clients upon request from the PS. In recent years, to reduce the waiting time and bandwidth demand between the main multimedia server (MMS) and a proxy server (PS), a number of caching and buffering techniques have been investigated. Most of these techniques use proxy servers with large storage space for caching videos which are requested frequently and serving the requests for these videos. The cached video is used to serve the future requests and only the uncached portions of the video are downloaded from the main multimedia server [1, 2 and12]. Proxy servers have been widely used in web contents to decrease the response time and alleviate the load of overloaded web servers. The same concept can be used for video streams in multimedia applications. Originally, a central video server provides a list of pre-recorded videos and delivers video contents to the user. A video proxy server (proxy server or proxy) residing at the same local area of the client can assist the delivery by taking advantage of its storage and proximity to the client. The proxy server stores complete or portion of a video. On the other hand, the proxy server transmits cached data to the client through LAN using its abundant and less expensive bandwidth. The motivation for our study for end-to-end video delivery is to reduce WAN bandwidth requirement [3, 5, and 12].

A proxy server can be considered as an intermediate node along the server-to-client path. This in effect will partition the server-to-client path into a server-to-proxy path and a proxy-to-client path. When a video is accessed by a client, the whole video or a portion of the video may have already been cached in the proxy server. If the video is not stored in the proxy server, it needs to be accessed from the central server and delivered to the client. It is possible at the same time the video will be cached in the proxy server to improve the future accesses [2]

Transmission and storage of MPEG videos requires high network bandwidth, which is one of the most expensive resources in Video on demand system. Therefore the critical part of video on demand is to optimize network bandwidth [7].

In this paper, we focus on developing an efficient algorithm to allocate and then reallocate the buffer at the proxy server, so that more number of videos can be stored at proxy server which in turn reduces the load of the main multimedia server, decreases the bandwidth requirement

Manuscript received November 5, 2008
Manuscript revised November 20, 2008

between central multimedia server and proxy server, client's waiting time.

2. Related Work

Proxy servers have been widely used in multimedia applications to decrease the waiting time and to reduce the load of the main multimedia server. Recent works investigate the advantages of connected proxy servers within the same intranet [3, 4 and 8].

In [3], Tay and pang has proposed an algorithm called GWQ (Global waiting queue) which reduces the client waiting time by sharing the load in a distributed loosely coupled VoD system. They have given a load sharing algorithm which balances the load between the heavily loaded proxy servers and lightly loaded proxy servers in a distributed VoD. They have assumed that

- Videos are replicated in all the servers
- Videos are evenly distributed in all proxy servers.

These assumptions require that the storage capacity of individual server should be very large to store all the videos.

Later in [4], Sonia Gonzalez, Navarro, Zapata proposed a more realistic algorithm to share the load in a distributed VoD system. In their research, they have demonstrated that their algorithm maintains a small waiting times using less storage capacity servers by allowing partial replication of the videos. They store the locally requested videos in each server. Depending on the user requests, the popularity of the videos and percentage of replication is determined.

In [6], we have proposed a unique VoD architecture and an efficient load sharing algorithm for distributed VoD system. The architecture consists of a main multimedia server (MMS) which is connected to a group of trackers (proxy server). Each tracker is in turn connected to a set of local proxy servers and these proxy servers are assumed to be interconnected in a ring fashion [LPSG].

Tracker maintains a database that contains complete information of the videos present in each proxy server of LPSG and manages the streaming of the requested video to the client through a particular proxy server(PS) by searching for the requested video among the other proxy servers of LPSG, if it is not present in the proxy server(PS) to which the client requesting for the video is connected.

And the proposed algorithm caches the video content at the proxy server currently requested by its users, by allocating required amount of buffer to all the frames of the requested video if available, otherwise dynamic buffer reallocation has been done to allocate the minimum buffer required to stream the video based on the popularity of the videos, reducing the client waiting time, and the bandwidth requirement from main centralized multimedia server to the client with the help of the tracker.

In this paper we propose a more efficient algorithm over the previous paper [6] for the architecture using frame differencing technique, in which, we take an advantage of similarity between successive video frames, If two successive frames have the same background, there is no need to store the background again. Instead, only the differences between the two frames need be stored. So this scheme allows PS to store and stream more number of videos from the proxy server, reducing the client waiting time and the bandwidth requirement between main multimedia server and the proxy server. The rest of the paper is organized as follows: section 3 presents the proposed approach and algorithm in detail, Section 4 presents the simulation model, Section 5 presents the simulation results and discussion, Section 6 finally concludes the paper and further work.

3. Proposed Architecture and Algorithm

3.1 Proposed Architecture

The proposed VoD architecture is as shown below in Fig.1. The tracker [TR] is a coordinating proxy server, which maintains a database that contains the information of the videos present in each proxy server of that LPSG [Local Proxy Servers Group]. The MMS, the tracker and the proxy servers of LPSG are assumed to be interconnected through optic fiber cables.

Initially, the total number of videos N is stored in the MMS. The distribution of these N videos among M proxy servers in a LPSG is done as follows. First, all the N videos are arranged with respect to their popularity. The popularity of a video in a server is defined as the frequency of requests to the video by the clients. The number of videos to be stored in all the proxy servers, k is estimated. The first k videos are selected from the popularity based sorted list and stored in each proxy server. The remaining N-k videos are evenly distributed among M proxy servers in LPSG.

When a request for a particular video arrives at a particular proxy server, the following three cases may happen:

- The requested video may be already present in that proxy server.
- The requested video may not be present in that proxy server, but it may be present in other proxy servers (one or more)of LPSG.

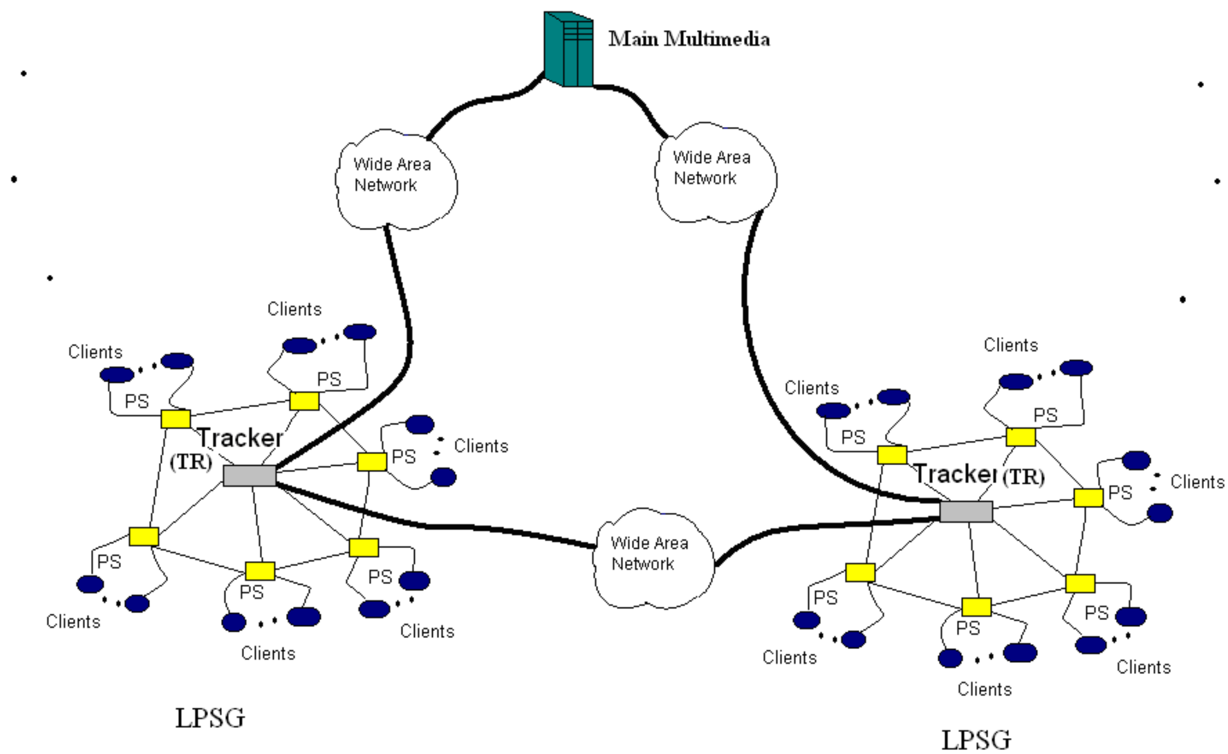

Fig.1

- The requested video may not be present in that proxy server, and also it may not be present in any of the other proxy servers of LPSG. In this case the requested video must be downloaded and streamed from the MMS.

If the requested video is present in proxy server at which request has arrived (PS), then the streaming of the video starts immediately and hence the client waiting time is very low.

If the requested video is not present in that proxy server, then the same is intimated to the tracker. The tracker checks whether the requested video is present in any of the other proxy server in that LPSG. If the requested video is present in one or more of the proxy servers in that LPSG, then the tracker checks whether the proxy server in which the video was found is neighbor to the requested proxy server.

If so, the tracker intimates the same to PS. Now the PS checks for the buffer availability. If sufficient buffer based on popularity is available, the streaming of video is initiated from the selected neighboring PS to the client. Hence the client waiting time is relatively higher. If required buffer based on popularity is not available, then whatever remaining buffer is available is allocated to requested video. If this allocated buffer is less than minimum buffer required to stream the video, then on-line dynamic buffer allocation using frame differencing technique will be done as follows:

[On-line dynamic Buffer Allocation with frame differencing Technique[ODBA+FDT]: We find the lowest popular video at the proxy server, which is completely or partially present in the PS and is currently not being streamed(off-line) and is allocated more than minimum buffer. If the popularity of the requested video is greater than the popularity of lowest popular video having more than minimum buffer, then the buffer except the minimum buffer is deallocated from the lowest popular video and this deallocated buffer is allocated to the newly requested video. On the other hand if the popularity of the requested video is less than the popularity of the lowest popular video having more than minimum buffer, then the minimum buffer is deallocated from lowest popular video and this deallocated buffer is allocated to the newly requested video with condition that the lowest popular video still having the minimum buffer. If we do not find the lowest popular off-line video having more than minimum buffer, then using **frame differencing technique(FDT)** (taking the advantage of similarity between successive video frames, If two successive frames have the same background, there is no need to store the background again. Instead, only the differences between the frames need be stored. Instead of describing every pixel in every frame, this technique describes all of the pixels in the first frame, and then for every frame that follows, describes only the pixels that are different from the previous frame. **.** If most of the pixels in the frame are different from the previous frame, then the scene identifier gives indication of the new scene and the requirement of describing the new frame, allowing for every pixel to be described or rendered. Each whole rendered frame is referred to as a **key frame**) reallocate the buffer of the lowest popular video, and deallocate the extra buffer leaving the minimum buffer required to stream that video based on popularity. Allocate this deallocated buffer to newly requested video, if it is minimum enough to stream the video. Otherwise find the lowest popular off-line video with minimum buffer which has been in the buffer for more than the threshold time, and is replaced by newly requested video irrespective of the popularity.**]**

Otherwise (if it not a neighbor and present in more than one proxy server of LPSG), the tracker selects one proxy server such that the streaming path from the selected proxy server to the requested proxy server is optimal. Then the tracker intimates the requested PS. Now the PS checks for the buffer availability. If sufficient buffer based on popularity is available, then the streaming of video is initiated from the selected PS of LPSG to the client. Hence the relative client waiting time is higher. If required buffer based on popularity is not available, then whatever remaining buffer is available is given to requested video. If this allocated buffer is less than minimum buffer required to stream the video, then ODBA+FDT will be done as explained above. Now the streaming is initiated from the selected PS of LPSG to the client through the requested PS.

If the requested video is not present in any of the proxy servers in that LPSG, then the tracker initiates the streaming of the requested video from the MMS to the requested proxy server and the same is intimated to the requested PS. Then the PS checks for the buffer availability. If sufficient buffer based on popularity is available, then the TR initiates streaming of video from the MMS to the client through the requested PS. If required buffer based on popularity is not available, then whatever remaining buffer is available is given to requested video. If this allocated buffer is less than minimum buffer required to stream the video, then ODBA+FDT will be done as explained above. Now TR initiates downloading and streaming of the requested video from the MMS to the client through PS. Whenever sufficient bandwidth and buffer is not available in the above operation, the user request is rejected, which is a rare possibility as shown by our simulation results.

3.2Proposed ODBA+FDT Algorithm

[Nomenclature:

Psi - Proxy Server i
TR -Tracker
MMS – Main multimedia Server
V_{req} - Requested Video
TR_{lpsg} - Local Proxy Servers Group Under the TR
ps(TR_{lpsg}) - Proxy server in which V_{req} is found in $TR_{lpsg,}$
BUFreq - Required buffer based on popularity
Wttm - waiting time in ms**]**

When a request for a video V (V_{req}) arrives at a particular time t to particular PS i under TR do the following:

```
If (V_req is present at PSi)   //Present at PSi
 {
   Stream the V_req to the user immediately from PSi.
   Wait(Wttm<=50ms)(According to Simulation results)
 }
else
 {
   Pass the request to the TR
   If (V_req Є TR_lpsg )
     If (ps(TR_lpsg ) is left or right neighbor to PSi)  //
                  Present at left or right neighboring PS
      {
        {
          If  (BUFreq is available at PSi)
            PSi allocates BUFreq to V_req &  initiates
            the streaming of the V_req  to the user
            from  ps(TR_lpsg ) through PSi.
        else
            ODBA+FDT is done  based on popularity
            depending on the availability of minimum
            buffer  required to stream the video. Then
            PSi initiates the streaming of the V_req to
           user  from   ps(TR_lpsg ) through PSi.
        }
         Wait(wttm<=150ms)
      }
     else                //Present in other  PSs of LPSG

      {
        {
```

```
        If  (BUFreq is available at PSi)
            PSi allocates BUFreq to V_req & TR
            initiates the  Streaming of the V_req to the
            user through PS from  ps(TR_lpsg ) using
            the optimal path  given by TR
        else
            DBA+FDT is done  based on popularity
            depending on the availability of minimum
            buffer  required to stream the video. Then
            PSi initiates the streaming of the V_req to
            user from ps(TR_lpsg ) using the optimal
            path given by TR through PSi
        }
        Wait(wttm<=250ms)
      }
  else        // Not present in any of the PSs of LPSG
   {
     {
       If  (BUFreq is available at PSi)
          PSi allocates BUFreq to V_req &  initiates the
          downloading of the V_req  from MMS to  the
          user through PSi
       else
          ODBA+FDT is done  based on popularity
          depending on the availability of minimum
          buffer  required to stream the video. Then PS
          initiates the streaming of the V_req to user
          from ps(TR_lpsg ) through Psi
       }
    wttm<=450
   }
If sufficient buffer can not be allocated
   Reject the request  for V_req

}
```

4. Simulation Model

Our simulation model consists of a single Main multimedia server and a set of trackers. Each tracker is connected to a group of proxy servers which are interconnected among themselves in ring fashion through fiber optic cables. The following are the assumptions made in our simulation model. The user requests for the video follows Zipf law of distribution. The sizes of the videos are uniformly distributed over a range, the number of proxy servers in each group is assumed to be same.

The performance parameters are load sharing among the proxy servers of LPSG, the reductions of the load on the main multimedia server and client waiting time for the requested videos.

5. Results

The results presented below are an average of several simulations conducted on the model for a particular proxy server PSi of a specific LPSG. The values considered for simulation are as follows. Sizes of videos are 350MB to 550MB, number of proxy servers in each group is 6, each simulation is carried out for 1500 seconds.

Consider Fig.2, which shows, the number of videos served from PSi using On-line dynamic buffer allocation with frame differencing technique (ODBA+FDT) are more when compared to the number of videos served from PSi with On-line dynamic buffer allocation with out considering frame differencing technique(ODBA-FDT).

Fig.3 shows the number of videos served from NBRPSs [Neighboring proxy servers to PSi] using (ODBA+FDT) are less when compared to the number of videos served from NBRPSs (ODBA-FDT).

Fig.4 for number of videos served from LPSG-NBRPSs [proxy servers in LPSG other than neighboring proxy servers] using (ODBA+FDT) are less when compared to the number of videos served from LPSG-NBRPSs using (ODBA-FDT)

Fig.5 shows the number of videos downloaded from MMS [Main Multimedia Server] using (ODBA+FDT) are very less when compared to the number of videos downloaded from MMS using (ODBA-FDT) because as we stream through PSi our proposed scheme buffers more number of frequently requested(popular) videos at PSi by allocating and then reallocating the buffer using (ODBA+FDT). So later requests for those videos can be served from PSi

The videos that are frequently requested by the clients are stored in local proxy server (PSi). When there is a request for these videos, streaming starts from the PSi immediately and hence the waiting time for these videos is very less as shown in Fig.2, Fig.6 and Fig.11.

If the requested videos are present in the NBRPSs in that LPSG, then the streaming of requested video is initiated by the tracker from NBRPSs to the requested proxy server, so the waiting time for these videos is very small as shown in Fig.3, Fig.7 and Fig.11.

Otherwise also, more number of videos are served from LPSG-NBRPSs through PSi, which reduces frequent downloading of requested videos from MMS to the PSi which in turn reduces the waiting time of the requested videos which are not present at PSi as shown in Fig.4, Fig.8 and Fig.11.

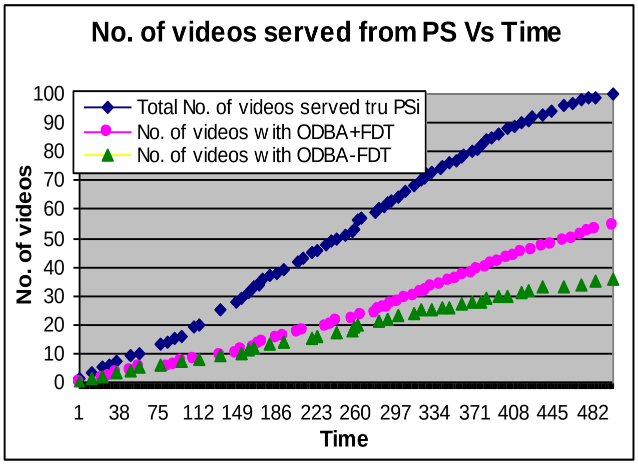

Fig.2

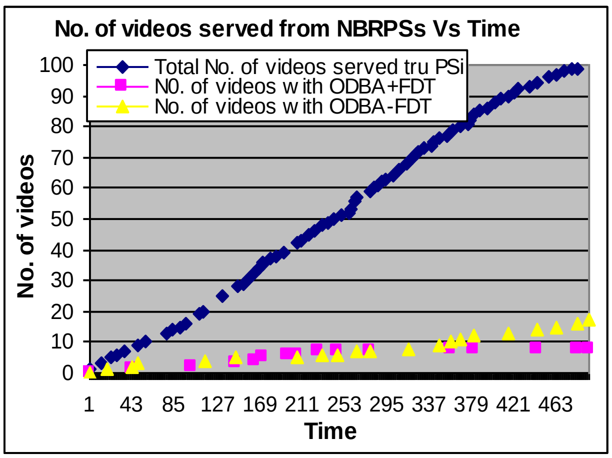

Fig.3

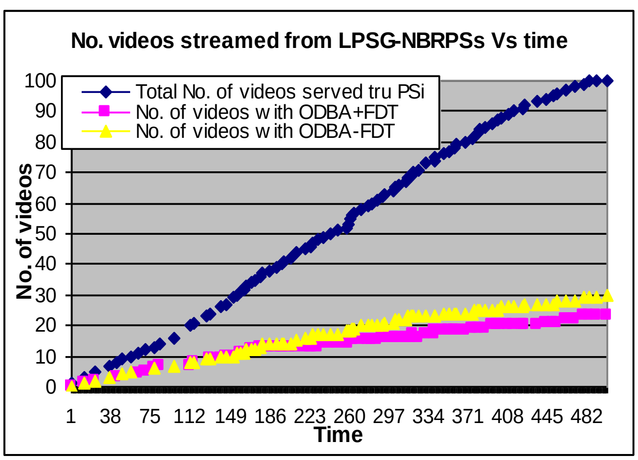

Fig.4

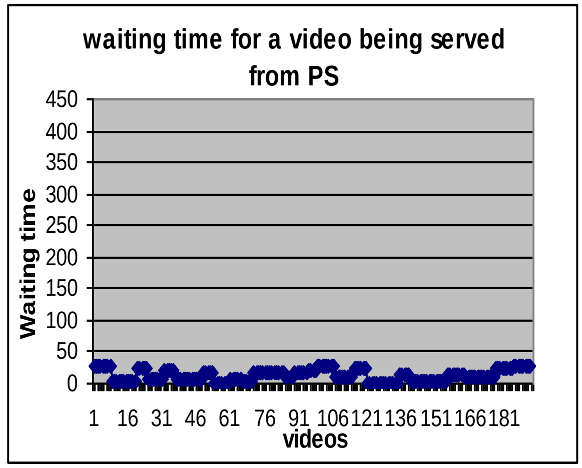

Fig.6

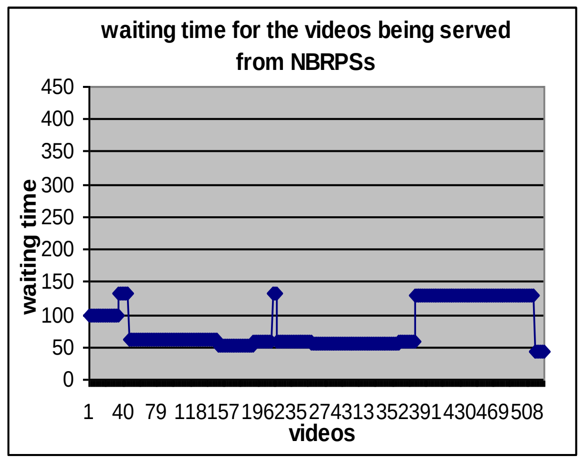

Fig.7

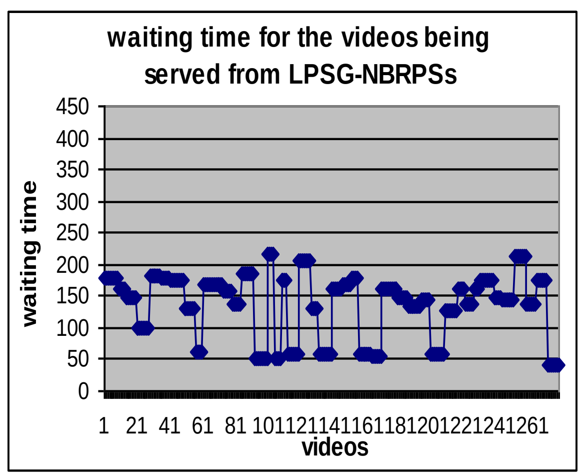

Fig.8

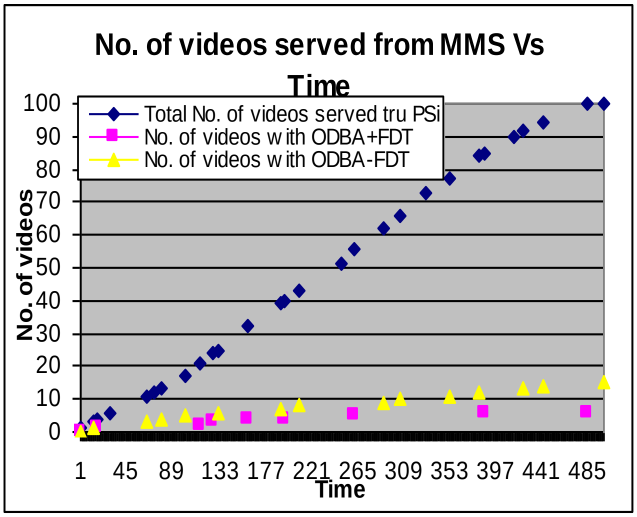

Fig.5

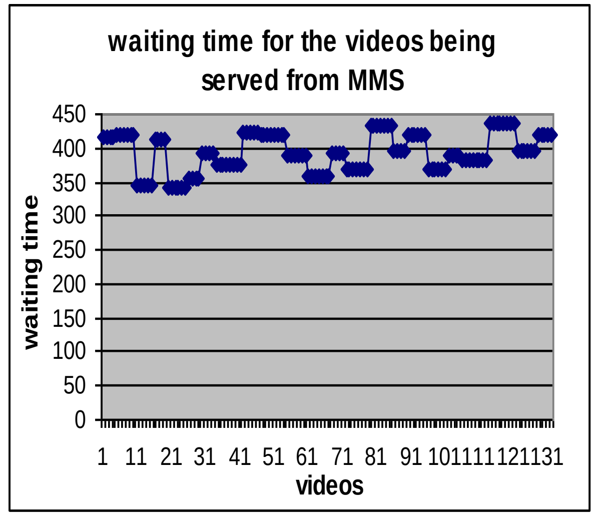

Fig.9

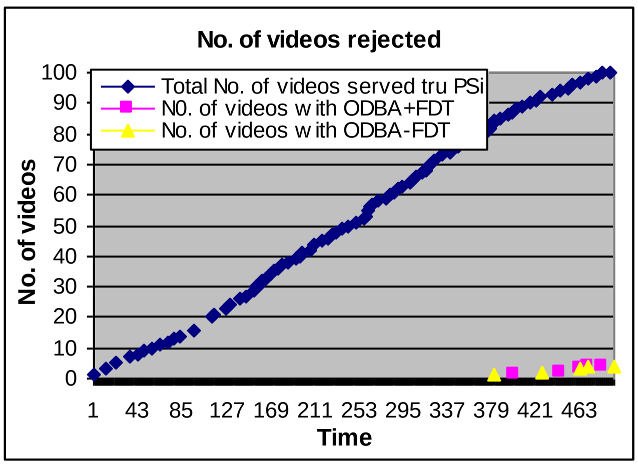

Fig.10

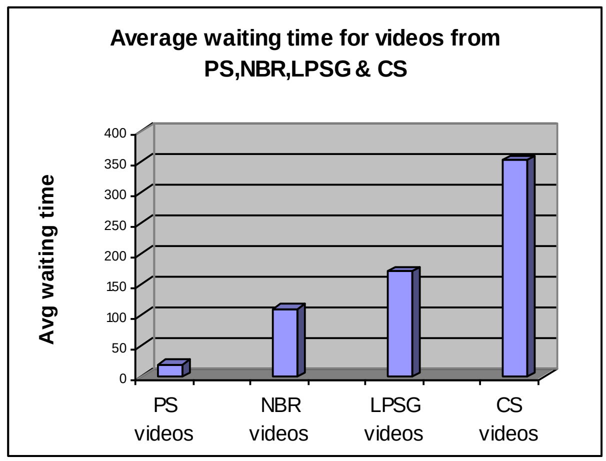

Fig.11

As shown in Fig.5, Fig.9 and Fig.11, very few numbers of videos are downloaded from MMS and streamed through PSi, which are not present at PSi and LPSG. Even though the waiting time seems to be more it is acceptable because very few number of videos are taken from MMS. Fig.10 shows very few requests are rejected

6. Conclusion

In this paper we have proposed a unique VOD architecture and an efficient buffer allocation algorithm using On-line dynamic buffer allocation with frame differencing technique (DBA+FDT). Though the proposed new algorithm causes a minor increase in time in applying frame differencing technique, As it increases the storage of number of videos at PSi by allocating and then reallocating the buffer for the videos that are streamed through PSi using ODBA+FDT, the number of videos downloading from MMS has been reduced successfully, in turn reduces the bandwidth requirement between the MMS and the proxy servers and the client waiting time significantly. And simulation shows promising results. The proposed architecture and algorithm reduces the bandwidth requirement between the main multimedia server and the proxy servers, maintains high QoS for the users with minimum waiting time. The future work is being carried out to investigate efficient methods for bandwidth management considering the popularity of the videos.

References:

[1] H S Guruprasad, R Ashok kumar, H D Maheshappa Ganesan, “Mobile Agent Based Efficient Channel Allocation For VoD”, *IASTED International Conference on Communication Systems & Applications [CSA 2006*] 3rd -5th July, 2006, Banff, CANADA

[2] H S Guruprasad, M Dakshayini , H D Maheshappa & P Geethavani, “Efficient Buffer Allocation For Video-On-Demand System Based On Popularity”, *1st National Conference on Computing Technologies*,3 March, 2006, Sivakasi, pp 31-35.

[3] Y.C Tay and HweeHwa Pang, “Load Sharing in Distributed Multimedia-On-Demand Systems”, *IEEE Transactions on Knowledge and data Engineering*, Vol.12, No.3, May/June 2000.

[4] S. González, A. Navarro, J. López and E.L. Zapata, "Load Sharing in Distributed VoD (Video on Demand) Systems". *Int'l Conf. on Advances in Infrastructure for e-Business, e-Education, e-Science, and e-Medicine on the Internet (SSGRR 2002w***)**, L'Aquila, Italy, January 21-27, 2002.

[5]Yuewei Wang, Zhi-Li Zhang, David H.C. Du, and Dongli Su “A Network-Conscious Approach to End-to-End Video Delivery over Wide Area Networks Using Proxy Servers”, *IEEE INFOCOM*, pp 660-667, April 1998.

[6]M Dakshayini,H S Guruprasad,H D Maheshappa,A S Manjunath,“Load Balancing in Distributed VoD using Local Proxy Server Group [LPSG]”, iInt’l Conf. on “Computational Intelligence And Multimedia Applications 2007 (ICCIMA’07), Dec 2007, Mepco Schlenk Engineering College, Sivakasi, Tamilnadu, India.

[7]Avanapong, W. Tran, M. Zhou, J. Krishnamohan. S, “Video caching network for on-demand video streaming”, *GLOBECOM’02 IEEE Global Telecommunications*, 17-21 Nov. 2002, pp 1723- 1727 vol.2.

[8]S. Acharya and B. C. Smith, “Middleman: A video caching proxy server,” in *Proc. of NOSSDAV*, June 2000.

[9]Y.-W. Park, K.-H. Baek and K.-D. Chung, “Reducing network traffic using two-layered cache servers for continuous media data on the internet”, in *Proc. of the IEEE Int’l. Conf. on Computer Software and Applications*, 2000, pp. 389–374.

[10] S. Paknikar, M. Kankanhalli, K. R. Ramakrishnan,S. H. Srinivasan, and L. H. Ngoh, “A caching and streaming framework for multimedia,” in *Proc. of ACM Multimedia’2000*, CA, October 2000, pp. 13–20.

[11] N.Ansari, H.Liu, Y.Q.Shi and H.Zhao “Dynamic Bandwidth Allocation for VBR Video Transmission”, Journal of Computing and Information Technology-CIT 11, 2003,4,309-317

[12] Wei-hsiu Ma and David H.C.Du, fellow “Reducing Bandwidth Requirement for Delivering Video Over Wide Area Networks With Proxy Server” IEEE TRANSACTIONS ON MULTIMEDIA, VOL.4, NO.4,DEC 2002.