Dynamic Load Balancing Architecture for Distributed VoD using Agent Technology

H S Guruprasad hs_gurup@yahoo.com
Research Scholar, Dr MGR University
Asst Prof& HOD / Dept of ISE
BMSCE, Bangalore, India

Dr. H D Maheshappa hdmappa@gmail.com
Prof & HOD / Dept of E & C
RevaITM
Bangalore, India

Abstract

This paper proposes a load balancing algorithm for a distributed VoD architecture using agents. A mobile agent is used to frequently update the popularity of the videos based on which channel allocation is done effectively. The proposed approach groups a set of local proxy servers into a Local Proxy Server Group [LPG] for load balancing among the proxy servers, to reduce the load on the central multimedia server, to reduce storage redundancy among the proxy servers and to maximize the channel utilization. The simulation results prove the load balancing among the local proxy servers, reduction of load on central multimedia server, maximum channel utilization and more channel allocation for popular videos.

Keywords: Load Balancing, Mobile Agent, Channel Allocation, Waiting Time, Popular Videos, Proxy Server

1. INTRODUCTION

Agents are autonomous programs which can understand an environment, take actions depending upon the current status of the environment using its knowledge base and also learn so as to act in the future. Autonomy, reactive, proactive and temporally continuous are mandatory properties of an agent. The other important properties are commutative, mobile, learning and dependable. These properties make an agent different from other programs. The agents can move around in a heterogeneous network to accomplish their assigned tasks. The mobile code should be independent of the platform so that it can execute at any remote host in a heterogeneous network [1, 3, 7].

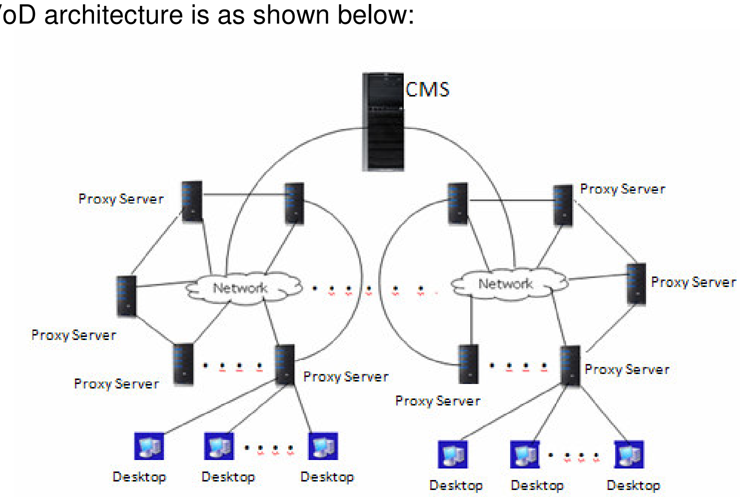
A video-on-demand system can be designed using any of the 3 major network configurations – centralized, networked and distributed. In a centralized system configuration, all the clients are connected to one central server which stores all the videos. All the client requests are satisfied by this central server. In a network system configuration, many video servers exist within the network. Each video server is connected to a small set of clients and this video server manages a subset of the videos. In a distributed system configuration, there is a central server which stores all the videos and smaller servers are located near the network edges. When a client requests a particular video, the video server responsible for the requests ensures continuous playback for the video [2].

Proxy servers are widely used in multimedia networks to reduce the load on the central server and to serve the client requests faster. In [4], Tay and Pang have proposed an algorithm called GWQ [Global Waiting Queue] which shares the load in a distributed VoD system and hence reduces the waiting time for the client requests. This load sharing algorithm balances the load between heavily loaded proxy servers and lightly loaded proxy servers in a distributed VoD. They assumed that videos are replicated in all the servers and videos are evenly required, which requires very large storage capacity in the individual servers. In [6], Sonia Gonzalez, Navarro, Zapata proposed a more realistic algorithm for load sharing in a distributed VoD system. Their algorithm maintains small waiting times using less storage capacity servers by allowing partial replication of videos. The percentage of replication is determined by the popularity of the videos.

In this paper, we propose a new load balancing algorithm and VoD architecture for distributed VoD system. This architecture consists of a Central Multimedia Server [CMS]. A set of local Proxy servers are connected together in the form of a ring to form a Local Proxy Server Group [LPG]. All the LPG's are connected to the CMS. All connections are made through fiber optic cables. The rest of the paper is organized as follows: Section 2 presents the proposed architecture, section 3 presents the proposed algorithm, Section 4 presents the simulation model, Section 5 presents the simulation results and discussion, Section 6 finally concludes the paper and further work.

2. PROPOSED ARCHITECTURE

The proposed VoD architecture is as shown below:



A group of Proxy Servers are connected together in the form of a ring called Local Proxy Server Group [LPG]. A set of clients (users) are connected to each Proxy Server. This group of local Proxy Servers is connected to the Central Multimedia Server [CMS] through fiber optic cables. One of the Proxy Servers in the LPG acts as a coordinator and maintains a database which contains the information of the videos present in each Proxy Server in that LPG and also the popularity of the videos in that LPG. A mobile agent tours through the Proxy Servers of a LPG periodically to find and update the set of videos present in each Proxy Server of the LPG and the popularity of the videos in the LPG. This information is shared among all the Proxy Servers in the LPG.

Initially, all the N number of videos are stored in the CMS. The distribution of the videos is done as follows:

First, all the N videos are arranged based on their popularity. The popularity of a video is directly proportional to the number of hits for the video. The number of requests to a video follows Zipf law of distribution. We select the first m videos from the popularity based sorted list and stored in each proxy server. The remaining videos are stored depending on the local popularity in the proxy servers.

When a request for a video arrives at the PS, the following cases happen:

- The requested video is present at the PS[Proxy Server]
- The requested video is not present at the PS, but is present either in LPS[Left neighboring Proxy Server] or RPS[Right neighboring Proxy Server]
- The requested video is present in both LPS and RPS
- The requested video is not present in both LPS and RPS, but is present in one of the Proxy Servers in that LPG.
- The requested video is not present in any of the Proxy Servers in the LPG

If the requested video is present in the PS, then the real time transmission of the video starts immediately with the video content being streamed to the client from the PS. If the requested video is not present in the PS, then we check whether it is present in the LPS or in RPS.

If the requested video is present only in LPS, then we check the number of channels allocated for popular videos b/w LPS & PS and CMS & PS. If more numbers of channels are allocated for popular videos b/w LPS & PS, then path LPS-PS is selected, otherwise the path CMS-PS is selected. If the requested video is the first request, then all the channels are allocated for this video. Otherwise, depending on the popularity of the requested video, the channels are allocated as follows: If the requested video is more popular than the videos being streamed in the channels, then more number of channels are allocated for the requested video by deallocating channels from the lesser popular videos being streamed in the channels. Otherwise, appropriate numbers of channels are allocated depending on its popularity and the popularity of the videos streamed. Then the channel allocation of the other videos is dynamically adjusted, if needed.

If the requested video is present only in RPS, then we check the number of channels allocated for popular videos b/w RPS & PS and CMS & PS. If more numbers of channels are allocated for popular videos b/w RPS & PS, then path RPS-PS is selected, otherwise the path CMS-PS is selected. If the requested video is the first request, then all the channels are allocated for this video. Otherwise, the channel allocation is done in the same way as given above.

If the requested video is present in both RPS and LPS, then we check the number of channels allocated for popular videos b/w RPS & PS, LPS & PS and CMS & PS. Among these 3 paths, we select the path in which more number of channels are allocated for most popular videos. If the requested video is the first request, then all the channels are allocated for this video. Otherwise, the channel allocation is done in the same way as given above.

If the requested video is not present in both LPS and RPS, but is present in one of the Proxy Servers in that LPG, then appropriate number of channels are allocated within the LPG by finding the optimal path from the PS and the server in LPG having the requested video depending on the popularity of the requested video.

If the requested video is not present in any of the Proxy Servers in the LPG, Then the path PS-CMS is selected.

If the requested video is the first request, then all the channels are allocated for this video. Otherwise, the channel allocation is done in the same way as given above.

3. PROPOSED ALGORITHM

Nomenclature:
[PS: Proxy Server
LPS: Left neighboring Proxy Server
RPS: Right neighboring Proxy Server
NPS: Neighboring Proxy Servers
LPG: Local Proxy Server Group
CMS: Central Multimedia Server
NCAPV(x): Number of channels allocated for popular videos between PS and x]

```
Channel_allocation(y)
 {
   If (requested video is the first request)
        All the channels between PS and y are allocated to this video
   else
     {
      - Number of channels between PS and y are allocated depending on the available number
        of channels and proportional to the popularity of the requested video
      - Dynamically adjust the channel allocation for the other videos(if required)
     }
 }
```

When a request for a video m arrives at a particular time t, do the following:

```
If (requested video is present in PS)
        Start streaming from PS
else
 {
        If (requested video is present in only LPS)
         {
          If (NCAPV (LPS) >= NCAPV (CMS))
                Channel_allocation (LPS)
          else
                Channel_allocation (CMS)
         }
         else if (requested video is present in only RPS)
                {
                      If (NCAPV (RPS) > =NCAPV (CMS))
                          Channel_allocation (RPS)
                       else
                          Channel_allocation (CMS)
                }
        else If (requested video is present in LPS & RPS only)
              {
                 If (NCAPV (RPS) >= NCAPV (CMS) and (NCAPV (RPS) >=NCAPV (LPS))
                            Channel_allocation (RPS)
                 If (NCAPV (LPS) >= NCAPV (CMS) and (NCAPV (LPS) >=NCAPV (RPS))
                            Channel_allocation (RPS)
                   else
                            Channel_allocation (CMS)
              }
             else if (requested video is present in LPG – NPS's)
                    {
```

```
                    If ((NCAPV (LPG-NPS's)>= NCAPV (CMS))
                         Channel allocation is done within the LPG by finding the optimal path
                         from the PS and the server in LPG having the requested video
                  else
                            Channel_allocation (CMS)
                    }
                else
                            Channel_allocation (CMS)
}
```

4. SIMULATION MODEL

The simulation model consists of a single Central Multimedia Server [CMS], and a few proxy servers in one local proxy server Group [LPG]. The following are the assumptions made in the 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 channels between PS & LPS, between PS & RPS, between the Proxy Servers in a LPG and between PS & CMS are assumed to be same.

The performance parameters are load sharing among the proxy servers, reduction of load on the CMS and channel utilization between PS & LNPS, PS & RNPS and PS & CMS and between the Proxy Servers of a LPG.

5. RESULTS & DISCUSSION

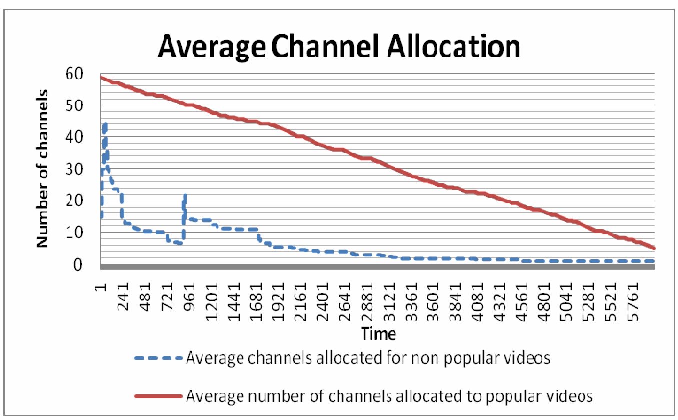


FIGURE 1: Average channel Allocation

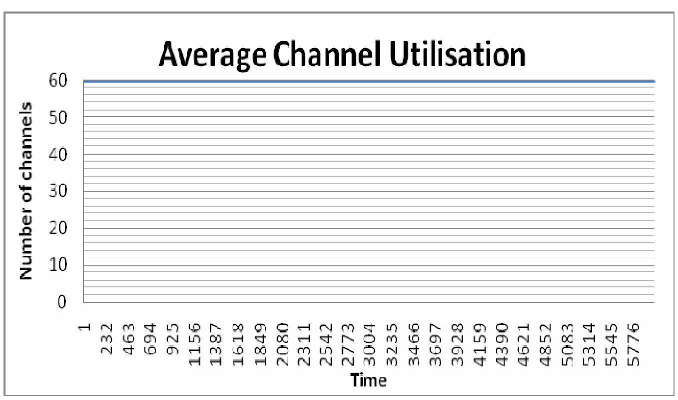


FIGURE 2: Average Channel Utilisation

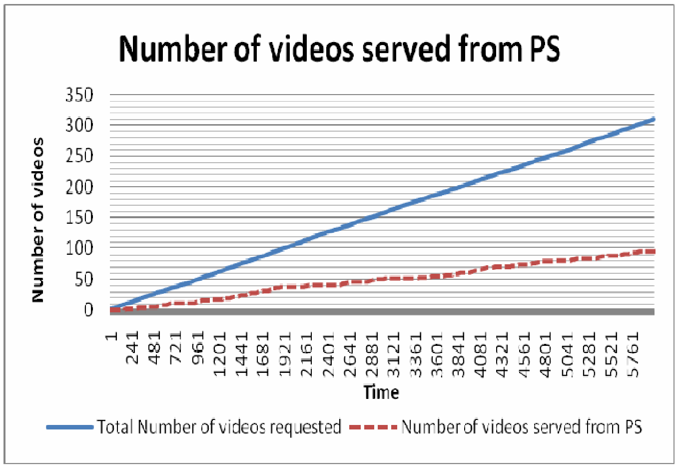


FIGURE 3: No. of videos served from PS

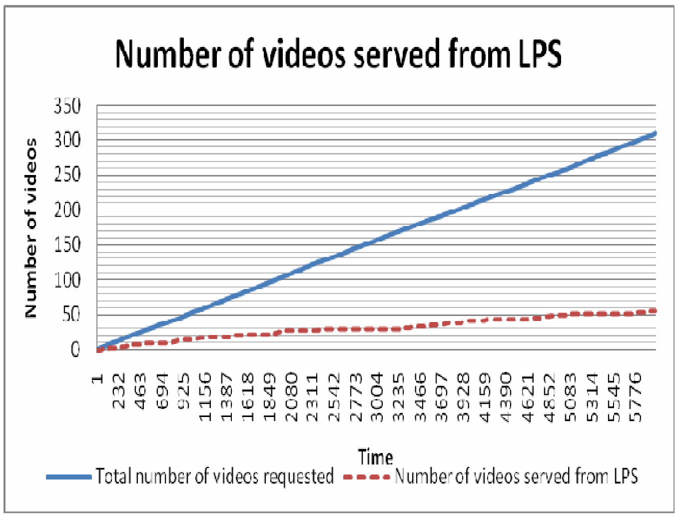


FIGURE 4: No. of videos served from LPS

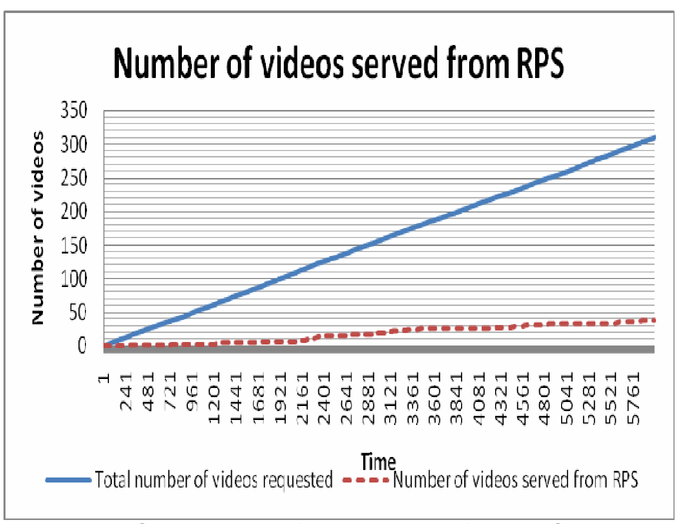


FIGURE 5: No. of videos served from RPS

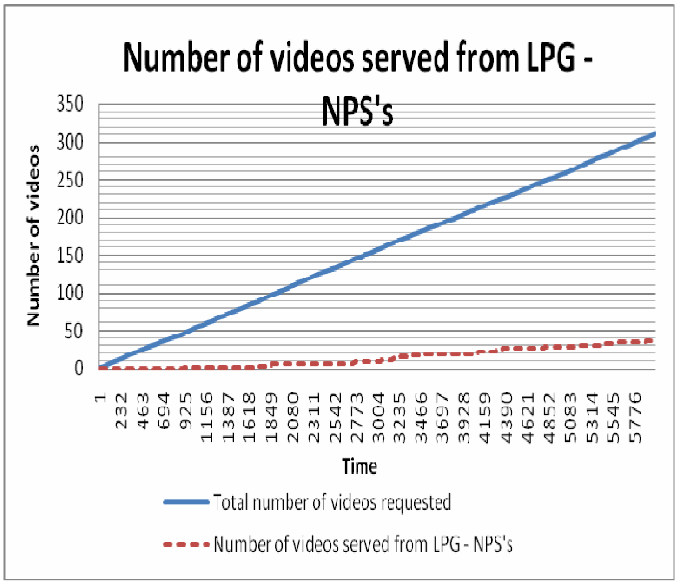


FIGURE 6: No. of videos served from LPG-NPS's

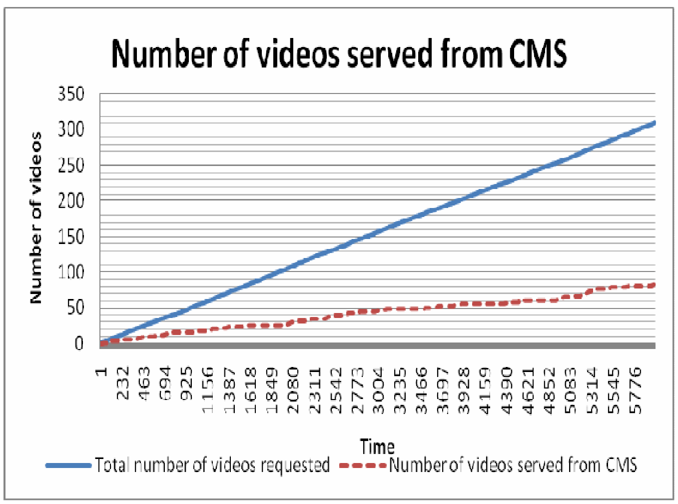


FIGURE 7: No. of videos served from CMS

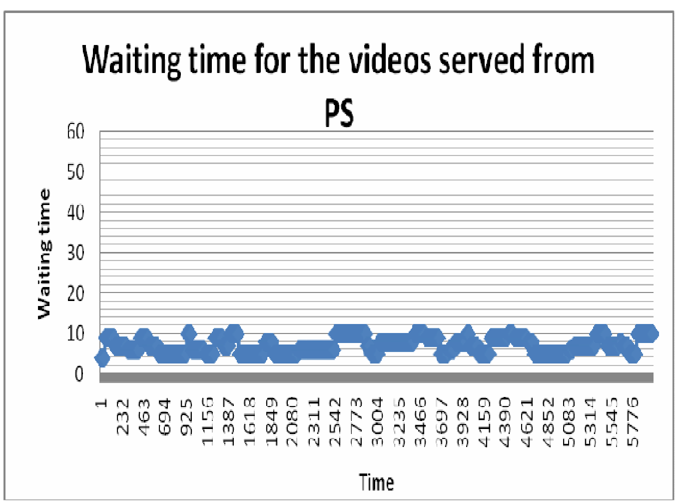


FIGURE 8: Waiting time for the videos served from PS

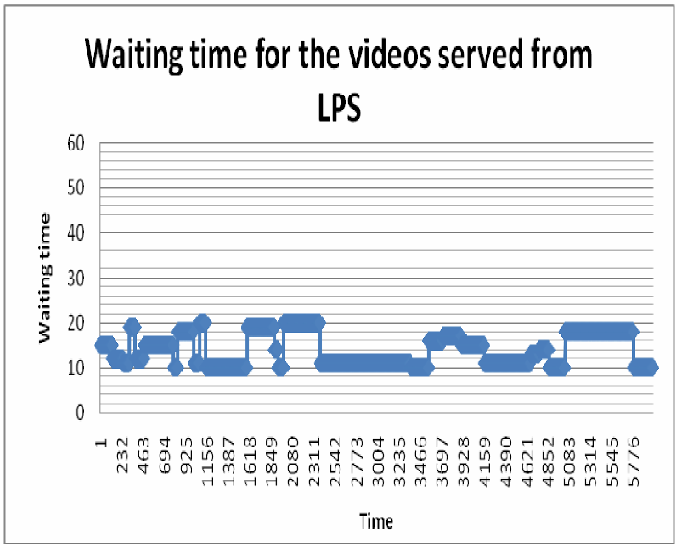


FIGURE 9: Waiting time for the videos served from LPS

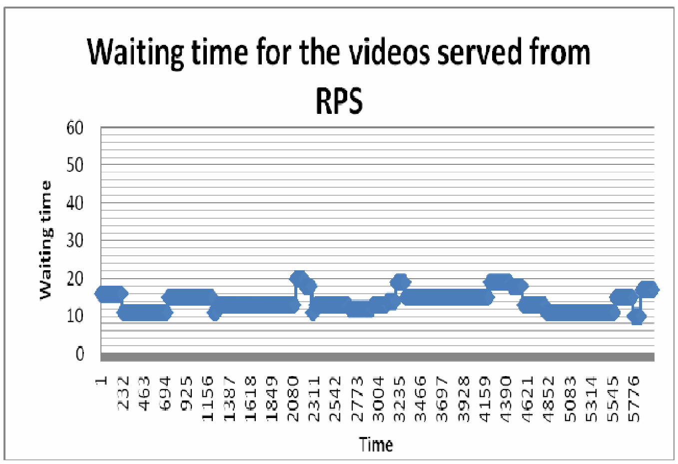


FIGURE 10: Waiting time for the videos served from RPS

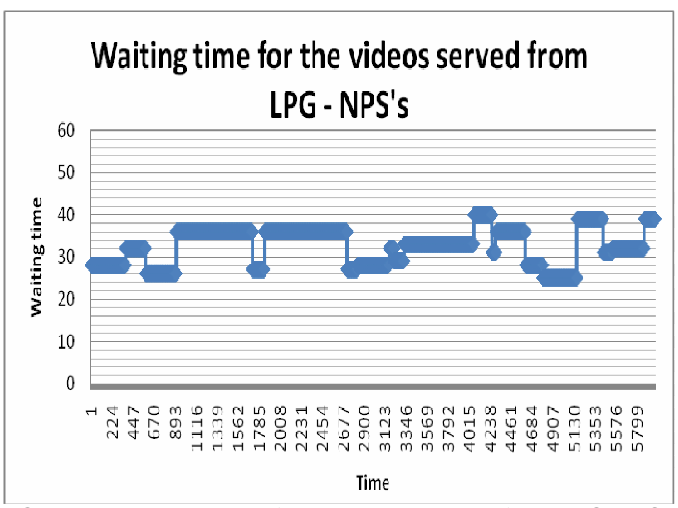


FIGURE 11: Waiting time for the videos served from LPG-NPS's

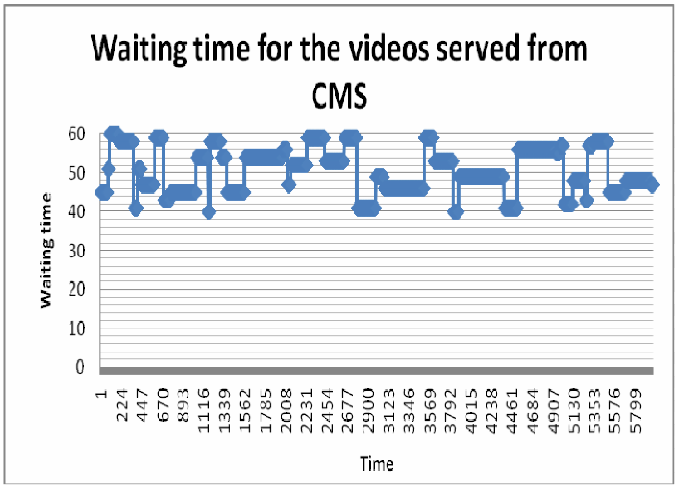


FIGURE 12: Waiting time for the videos served from CMS

The results presented are an average of several simulations conducted on the model. The sizes of the videos are taken in the range 300MB to 800MB. The number of proxy servers are considered in LPG is 6 and each simulation is carried out for 6000 seconds.

Fig 1 shows the average channel allocation for popular videos and non popular videos. Our channel allocation algorithm allocates more channels for popular videos than the non popular videos. The average channel utilization is always maximum because all the channels are allocated among the videos being streamed as shown in Fig 2.

The videos that are requested frequently are stored in the PS. When there is a request for these videos, streaming starts immediately and hence the waiting time for these videos is very less as shown in Fig 3 and Fig 8.

When the requested video is found in LPS or RPS, the streaming is initiated to the requested proxy server from the LPS or RPS and the waiting time is very small as shown in Fig 4, Fig 5, Fig 9 and Fig10.

When the requested video is found in some proxy servers other than LPS or RPS, the streaming is initiated to the requested proxy server and the waiting time is considerable as shown in Fig 6 and Fig 11.

When the requested video is not found in any of the proxy servers in the LPG, the video has to be streamed from the CMS and the waiting time is more as shown in Fig 7 and Fig12.

6. CONCLUSION

In this paper, we have concentrated on the load balancing among the proxy servers and central multimedia server using agents. The simulation shows promising results. The algorithm always uses maximum number of channels between the proxy servers in a LPG and also between the CMS and the proxy servers of a LPG by allocating more channels to the more popular videos. Further work is being carried out to investigate load balancing by optimally balancing the channels and the buffer.

REFERENCES

1. M Dakshayini, H S Guruprasad, H D Maheshappa, A S Manjunath, “Load Balancing in Distributed VoD using Local Proxy Server Group [LPSG]”, International Conference on

Computational Intelligence And Multimedia Applications 2007 (ICCIMA'07), Dec 13th -15th 2007, India.

2. Santosh Kulkarni "Bandwidth Efficient Video on Demand Algorithm (BEVA) " *10th International conference on Telecommunications*, Vol 2 pp 1335-1342, 2003

3. S S Manvi and P Venkataram, "Mobile Agent based online Bandwidth allocation Scheme in Multimedia Communications", *IEEE GLOBECOM 2001 Conference* USA

4. 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.

5. Meng Guo and Mostafa H. Ammar and Ellen W. Zegura, "Selecting among Replicated Batching Video-on-Demand Servers", *Proceedings of the 12th International Workshop on Network and Operating System Support for Digital Audio and Video*, pp 155—163, 2002

6. S. Gonzalez, A. Navarro, J. Lopez 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.

7. Mohammed A. M. Ibrahim, "Distributed Network Management with Secured Mobile Agent Support", *International Conference on Hybrid Information Technology (ICHIT'06)*, 2006

8. P R Rao, Prasanna H Bammigatti," Delegation in Role Based Access Control Model for Workflow Systems", IJCSS: International Journal of Computer Science and Security, Volume 2, Issue 2, 2008.

9. Frederic Thouin, Mark Coates, Dominic Goodwill, "Video-on-Demand Equipment Allocation," *Fifth IEEE International Symposium on Network Computing and Applications (NCA'06*), pp 103-110, 2006

10. Hongliang Yu, Dongdong Zheng, Ben Y. Zhao, Weimin Zheng, "Understanding User Behavior in Large-Scale Video-on-Demand Systems", *Proceedings of the 2006 EuroSys conference*, Volume 40 , Issue 4 (October 2006), PP: 333 - 344

11. A.M.Natarajan, C.Venkatesh, R.Asokan, "Ant Based Dynamic Source Routing Protocol to Support Multiple Quality of Service (QoS) Metrics in Mobile Ad Hoc Networks", IJCSS: International Journal of Computer Science and Security, Volume 2, Issue 3, 2008.

12. A Dan, D Sitaram and P Shabuddin, "Dynamic batching policies for an on demand video server ", *Multimedia Systems,* pp 51-58, 1996

13. Gonzalez, A. Navarro, J. Lopez and E.L. Zapata, "A case study of Load sharing based on popularity in Distributed VoD systems", IEEE Transactions on Multimedia, Vol 8, No. 6, December 2006

14. Jiang Yu, Chun Tung Chou, Xu Du, Tai Wang, "Internal popularity of streaming video and its implication on caching",20th International Conference on Advanced Information Networking and Applications(AINA '06), 2006

15. Kamaksi Prasad V2, N H Ayachit1, Santosh L Deshpande1, "The time efficient security for broadcast Networks", IJCSS: International Journal of Computer Science and Security, Volume 2, Issue 2, 2008.