IACSIT International Journal of Engineering and Technology, Vol.2, No.2, April 2010
ISSN: 1793-8236

Transmission Cost Reduction Strategy for VoD Applications using Stochastic Approaches

M. Dakshayini, Dr. T.R. Gopalakrishnan Nair

Abstract—**In Video-On-Demand system, the streaming of high quality videos consume a significant amount of communication network resources. Hence the network bandwidth requirement and the transmission cost are the two key parameters to be considered. As the cost of proxy server is decreasing and demand for reduced network usage and transmission cost is increasing day by day, newer architectures are explored, innovative schemes are arrived at. Proxy caching can be used as a vital technique to reduce the network usage and the transmission cost of the VoD system. In this paper, we present novel three layer architecture of interconnected proxy servers with main multimedia server and a Tracker to optimize the bandwidth requirement, transmission cost and the Service delay of the system. We also propose an efficient prefix caching and distribution algorithm with load sharing technique to achieve high video availability close to the client. The simulation results demonstrate that it achieves significantly reduced network usage between the main server and the proxy server, and hence this reduces the transmission cost when compared to the pure proxy -based caching without cooperation among the proxy servers.**

Index Terms— **Video Streaming, Proxy caching, video distribution, Load sharing, Transmission cost.**

I. Introduction

The tremendous growth of World Wide Web has resulted in an increase of bandwidth consumption throughout the internet. Hence the network bandwidth has become more scarce and precious. Proxy caching has been recognized as an effective technique to reduce network traffic. Caching is also an important mechanism for improving both the performance and operational cost of multimedia networks [10]. Recent web access patterns show frequent requests for a small number of popular objects at popular sites. So a popular video can be streamed to the same network link once per request. In the absence of caching, this approach results in server over load, network congestion, higher request-service delay, and even the higher possibility of rejection of a clients request. Caching the videos which has a high demand at the proxy servers solves all these problems by distributing the load across the network [3].

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 clients. Proxy cache attempt to improve performance of the overall network communication in three ways [9]:

1) Reduce the request-service delay associated with obtaining documents (because the proxy cache is typically close to the user).
2) Lower the network traffic (the documents served already are available to the user for next time so less load on the network)
3) Reduce the Network cost.

In recent years, to reduce the Communication network usage on central server to proxy server path and transmission cost, a number of caching and buffering techniques have been proposed. Most of these techniques use proxy servers with large storage space for caching videos which are requested frequently. The cached data is used to serve the future requests and only the uncached portions of the videos are downloaded from the Main server [2].

Proxy servers have been widely used for multimedia contents to decrease the startup delay and to reduce the load of the Main multimedia server. Recent works investigate the advantages of connected proxy servers as a group of proxy servers [3, 5 and 8].

The organization of rest of the paper is as follows: Section II describes the related work. In section III we present a proposed approach and algorithm in detail, In section IV we present a simulation model, Section V presents the simulation results and discussion, Finally, in section VI, we conclude the paper and refer to further work.

II. Related work

This section briefly discusses the previous work as follows, Tay and pang [3] have proposed an architecture of proxy servers and an algorithm called GWQ (Global waiting queue). This reduces the initial startup delay by balancing the load between the lightly loaded proxy servers and heavily loaded proxy servers in a distributed loosely coupled VoD system. This work of assigning and reassigning the jobs between heavily loaded and lightly loaded proxy servers may increase the network traffic. They have replicated the complete videos evenly in all the servers, for which the storage capacity of individual proxy server should be very large to store all the videos. Sonia Gonzalez, Navarro, Zapata [4] proposed a more realistic algorithm to distribute the load in a distributed VoD system. In their research, they have demonstrated that their algorithm maintains a small initial start up delay using less

This work is supported by Multimedia research group, RIIC, Bangalore. Manuscript received June 9, 2009.
M. Dakshayini is with B M S College of Engineering, Bangalore, India (phone: 09880676988; fax: 080-26614357; e-mail: shantha_dakshu@yahoo.co.in).
Dr. T. R. Gopalakrishnan Nair, director of Research and Industry Incubation Centre, DSI, Bangalore (e-mail: trgnair@yahoo.com).

storage capacity servers by allowing partial replication of the videos. They store the locally requested videos in each server. Our algorithm differs by optimal distribution of videos among the group of proxy servers. This utilizes the aggregate storage space of proxy server's group more efficiently by caching the partial prefix -I at proxy and prefix-II at tracker in proportion to popularity. S.-H. Gary Chan et al.[7] Considers the exchange of cached contents with the neighboring proxy server without any coordinator. Our work differs, in which we have considered a group of proxy servers with a coordinator (Tracker) to make the sharing of videos more efficient. Another approach to reduce the aggregated transmission cost has been discussed in [6] by caching the prefix and prefix of suffix at proxy and client respectively. Since the clients are not trustable, and can fail or may leave the network at any time without any notice, they have adopted an additional mechanism to verify the client and cached data at client, which increases the overhead of such verification. Both searching of video in the whole cluster of proxy servers, and the verification process increases the client's waiting time. In this paper, we present a novel three layer architecture of interconnected distributed proxy servers with a target to optimize the network usage and the transmission cost. This architecture consists of a Main multimedia server [MMS] at the first layer, which is very far away from the user and is connected to a set of trackers [TR] at the second layer. Each tracker is in turn connected to a group of proxy servers and these proxy servers are assumed to be interconnected in a ring pattern at third layer, this arrangement of cluster of proxy servers is called as Local proxy servers Group [LPSG|L_p]. Each of such LPSG, which is connected to MMS, is in turn connected to its left and right neighboring LPSG in a ring fashion through its tracker. We also propose an efficient prefix caching and distribution algorithm at LPSG based on popularity of the video. Load sharing algorithm for the proposed architecture is also proposed to share the videos present among other proxy servers of LPSG. This increases the availability of videos at LPSG close to client and decreases the communication with MMS and hence the network usage on MMS-PS path and transmission cost.

Arranging the group of proxy servers in the form of LPSG provides several advantages as follows.

- *Reduced transmission cost*: distribution of videos among the PSs of LPSG based on the popularity of the videos, and sharing of these videos among the PSs of LPSG enables the system to serve maximum number of requests from LPSG itself. This reduces the communication demand at MMS and therefore can achieve significant reduction in network usage and transmission cost of the system.
- *Increased aggregate storage space:* by distributing the large number of videos across the PSs and TR of LPSG, high cache hit rate can be achieved. For example, if 10 PSs within a LPSG managed 500 Mbytes each, total space available is 5 GB. 200 proxies of LPSG could store about 100 GB of movies.
- *Load reduction*: caching the videos at PSs of LPSG based on their frequency of requests, provides service to more number of clients, thus by passing downloading of the complete video from the main multimedia server. This reduces the load of MMS
- *Scalability*: by adding more number of PSs the capacity of the system can be expanded. Interconnected TRs increases the system throughput.
-

III. Stochastic based Model and Problem Formulation

Let N be a stochastic variable representing the group of videos and may take the different values (videos) V_i (i=1,2 . . N) and the probability of the video V_i being taken is $p(V_i)$, let the set of values $p(V_i)$ be the probability mass function. Since the variable must take one of the values, it follows that $\sum_{i=1}^{N} p(v_i) = 1$. So the estimation of the probability of requesting V_i video, is $p(V_i) = \frac{n_i}{I}$, where I is the total number of observations. A cumulative distribution function denoted as $P(V_i)$ is the function that gives the probability of a request (random variable's) being less than or equal to a given maximum value.

Let S_i be the size (duration in minutes) of i^{th} video V_i (i=1..N) with mean arrival rates $\lambda_1 \ldots \lambda_N$ respectively that are being streamed to the users using M proxy servers (PSs) of J LPSGs (L_p p=1..J).

A. Distribution Strategy

Each TR and PSq has a caching buffer C_{PS},C_{TR} respectively. These buffers are large enough to cache total P and B minutes of H and K number of videos respectively. Before caching at LPSG, each video is segmented into 3 parts. First W_1 minutes of each video V_i is referred to as prefix-1(pref-1)$_i$ of V_i and is cached at any one of an appropriate PS of LPSG, in which relatively there is a high frequency of accessing the video V_i. This enables the LPSG system to cache more number of video prefixes close to the client. Next W_2 minutes of video V_i is referred to as prefix-2 (pref-2)$_i$ of V_i and is cached at TR of LPSG.

i.e. $P = \sum_{i=1}^{H} (\text{pref} - 2)_i$ and $B = \sum_{i=1}^{K} (\text{pref} - 1)_i$ (1)

$$N = K \times q \quad (2)$$

N= Total number of videos at LPSG
K= Number of videos that can be cached at each PS
q = Number of PSs

Depending on the frequency of arrivals of user requests to any video, the popularity of the videos and size (W_1 and W_2) of (pref-1) and (pref-2) to be cached at PS and TR respectively is determined.

i.e.
$[W_1(V_i) = (\text{pref-1})_i \text{ and } W_2(V_i) = (\text{pref-2})_i] \propto n_i$

$$W_1(\text{pref} - 1)_i = x_i \times S_i \quad \text{where } 0 < x_i < 1 \quad (3)$$

$$W_2(\text{pref}-2)_i = x_i \times (S_i - (\text{pref}-1)_i) \quad (4)$$

where $0<x_i<1$

Where x_i is the frequency of occurrences of requests for video i. And n_i is the number of requests for video V_i. Let b_i be the available bandwidth for V_i between the proxy and Main multimedia Server. After requesting for a video V_i at PS_q, the streaming of that video V_i may cost for the user by

$$TCost_i^{PSq} = \left[TC(pref-1)_i^{ps_q-U}\right] + \left[TC(pref-2)_i^{(TR-PS_q-U)}\right] + \left[TC(S-(pref-1)-(pref-2))_i^{(CMS-TR-PS_q-U)}\right] \quad (5)$$

where i=1..N, q=1...M and TCost is same as TC

Where the left hand part of the equation in the above model represents the total transmission cost required to retrieve and stream the requested complete video to the client. The right hand part represents the transmission cost (TCost) required to retrieve and stream (pref-1) from PS ($TC(\text{pref-1})_i$), (pref-2) from TR ($TC(\text{pref-2})_i$), and the remaining suffix (S-(pref-1)-(pref-2)) from central multimedia server (CMS) ($TC(S\text{-(pref-1)-(pref-2)}_i$)to the user respectively.

As the number of requests for the video increases the size of pref-1 is also increases, which is available close to the client at LPSG. The greedy distribution technique increases the storage of number of video prefix among the PSs of LPSG increasing the video availability at LPSG. Thus more number of users are serviced with maximum portion of the video fro LPSG itself. This in turn reduces the communication demand at MMS and hence reducing the bandwidth requirement on MMS-PS path also the transmission cost $TCost_i^{PSq}$, which is a non-linear function.

B. Problem Formulation

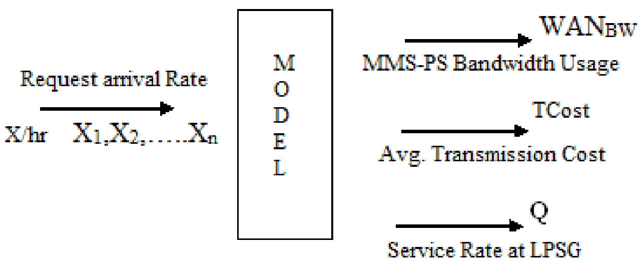

Figure 1: Simulation Model

The optimization problem is to maximize the service rate at LPSG and the portion of the video streamed from LPSG ((pref-1)+(pref-2)). Also to minimize the number of videos and portion of the video streamed from MMS ((S-(pref1)-(pref-2)) thereby minimizing the average transmission Cost for a video i at a PSq $TCost(V_i)^{PSq}$ as shown in Fig.1 can be formulated as follows:

Minimize Average Suffix transmission :

$$\sum_{i=1}^{N}\{S_i-(\text{pref}-2)_i-(\text{pref}-1)_i\} \quad (6)$$

$$\text{Minimize Average TCost :} \quad \left\{\sum_{j=1}^{Q}\sum_{i=1}^{N} V_i\right\} \Big/ Q \quad (7)$$

Where i = 1 …N,
q=1…M, and
Q- total number of requests
Subject to

- $C_{PS} = B = \sum_{i=1}^{K}(\text{pref}-1)_i$, (8)
- $C_{TR} = P = \sum_{i=1}^{H}(\text{pref}-2)_i$ (9)
- (pref-1) and (pref-2)>0 (10)

IV. Proposed VoD Architecture (LPSG)

The proposed 3 layer VoD architecture is as shown in Fig.2. This architecture consists of a MMS, which is connected to a group of trackers (TRs), Each TR has various modules like Interaction Module(IM) – to interact with the PS and MMS, Service Manager(SM_{TR}) – to handle the request from the PS, Database – to store the complete details of presence and size of (pref-1) of videos at at all the PSs, Video distributing Manager(VDM) – is responsible for deciding the videos and sizes of (pref-1), (pref-2) of videos to be cached, and also manages the distribution and management of these videos to group of PSs, based on video's global and local popularity as shown in the Fig. 3.

Each TR is in turn connected to a set of PSs. These PSs are connected among themselves in a ring fashion, Each PS has various modules like Interaction Module – to interact with the user and with the TR, Service Manager (SM_{PS}) – to handle the request from the user, Popularity agent(PA) – to observe and update the popularity of videos at PS as well as at TR, Cache Allocator(CA) – which allocates the Cache blocks based on video's popularity as shown in Fig. 3. To each of these PSs a large number of users are connected [LPSG]. Each proxy is called as a parent proxy to its clients. All these LPSGs are interconnected through their TR in a ring pattern. The PS caches the (pref-1) of videos distributed by VDM, and streams this cached portion of video to the client upon the request through LAN using its less expensive bandwidth. We assume that,

1. The TR is also a PS with high computational power and large storage compared to other proxy servers, to which clients are connected. It has various modules, using which it coordinates and maintains a database that contains the information of the presence of videos, and also size of (pref-1) and (pref-2) of video in each PS and TR respectively.
2. Proxies and their clients are closely located with relatively low communication cost[ ]. The Main server in which all the videos completely stored is placed far away from LPSG, which involves high cost remote communication.
3. The MMS, the TR and the PSs of LPSG are assumed to be interconnected through high capacity optic fiber cables.

In the beginning, all the N videos are stored in the MMS. The distribution of these N videos among M PSs in a LPSG is done by VDM as follows. First, all the N videos are arranged with respect to their popularity at jth LPSG. The popularity of a video is defined as the frequency of requests to this video per threshold time t by the clients. Here, we assume that the frequency of requests to a video follows Zipf law of distribution. Since the storage cache space of both proxy (C^{PS}), and TR (C^{TR}) is limited, the Video distribution module of the TR first executes the decision making algorithm of equation (3) and (4). Using this it will fix up the sizes (segments) of (pref-1) and (pref-2) of videos to be cached at PS and in its cache C^{TR} respectively, based on the frequency

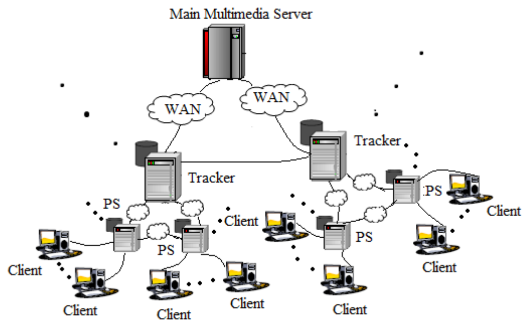

Figure 2: Distributed VoD Architecture (LPSG)

TABLE 1: LOAD SHARING ALGORITHM

[Nomenclature:
$TCost_{Vreq}$ -Transmission Cost required to stream V_{req}
P-1 - Prefix-1
P-2 - Prefix-2
p-u - Proxy server to User
t-p - Tracker to Proxy server
s-t - Main Server to Tracker]

When there is a request for a video v_i at a particular proxy PS_q of L_p, do the following:

If (V_{req} € PS_q)
{
(pref-1)$_{Vreq}$ is streamed immediately to the user

$$TCost_{Vreq} = TCost(p-1)^{p\text{-}u}_{Vreq} + TCost(p-2)^{(t\text{-}p)+(p\text{-}u)}_{Vreq} + TCost(S-(p-2)-(p-1)^{(s\text{-}t)+(t\text{-}p)+(p\text{-}u)}_{Vreq}$$

}
else - pass the request to the TR(Lp)
{
if (V_{req} € PS(Lp))
{
if(PS(Lp) is left or right NBR(PS_q)
SM_{TR} streams (pref-1)$_{Vreq}$ from NBR(ps_q), (pref-2) $_{Vreq}$ from its cache and the remaining portion from MMS

$$TCost_{Vreq} = TCost(p-1)^{(p\text{-}p)+(p\text{-}u)}_{Vreq} + TCost(p-2)^{(t\text{-}p)+(p\text{-}u)}_{Vreq} + TCost(S-(p-2)-(p-1))^{(s\text{-}t)+(t\text{-}p)+(p\text{-}u)}_{Vreq}$$

else
SM_{TR} streams the (pref-1)$_{Vreq}$ from OTR(PS_q), (pref-2) $_{Vreq}$ from its cache and the remaining portion from MMS to-User thru ps_q using optimal path found

$$TCost_{Vreq} = TCost(p-1)^{(p\text{-}p)+(p\text{-}u)}_{Vreq} + TCost(p-2)^{(t\text{-}p)+(p\text{-}u)}_{Vreq} + TCost(S-(p-2)-(p-1))^{(s\text{-}t)+(t\text{-}p)+(p\text{-}u)}_{Vreq}$$

}
else
{
Pass the request to left or right TR(NBR(Lp))

if (V_{req} € NBR(Lp))
TR(NBR(Lp)) streams the V_{req} from NBR(Lp)-user thru TR(Lp)

$$TCost_{Vreq} = TCost((p-1)+(p-2))^{(t-t)+(t-p)+(p-u)}_{Vreq} + TCost(S-(p-2)-(p-1)^{(s\text{-}t)+(t\text{-}p)+(p\text{-}u)}_{Vreq}$$

else
TR(Lp) downloads the complete V_{req} from MMS and streams to the user

$$TCost_{Vreq} = TCost(S)^{(s\text{-}t)+(t\text{-}p)+(p\text{-}u)}_{Vreq}$$

}
}

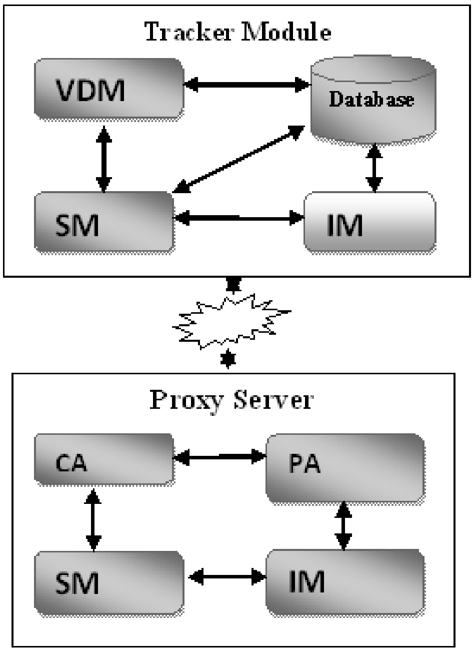

Figure 3: Modules of Proxy Server and Tracker

of user requests, in such a way that for most frequently accessing videos the total size of prefix-1(pref-1) at C^{PS} PS and prefix-2(pref-2) at C^{TR} should be more than the size of the rest of the portion of the video i.e.

$$((\text{pref}-1)+(\text{pref}-2))_i > (S-(\text{pref}-1)-(\text{pref}-2))_i \quad (11)$$

The corresponding entry is updated in its database at TR.

V. Proposed Load Sharing Algorithm

Whenever a client at PS_q wishes to play a video V_i, it first sends a request to its parent proxy PS_q, the SM_{PSq} immediately starts streaming the (pref-1) of video requested to the client, if it is present in its cache. Then it notifies the SM_{TR} to start the streaming of (pref-2) of V_i, then the interaction module of TR coordinate with CMS to download the remaining $(S\text{-(pref-1)-(pref-2)})_{Vi}$ portion. So only for $(S\text{-(pref-1)-(pref-2)})_{Vi}$ portion of video CMS is contacted, which may have to use the network bandwidth to CMS from TR of LPSG. Thus the transmission cost and initial service delay is almost negligible.

If it is not present in its cache, the IM_{PSq} forwards the request to its parent TR, VDM at TR searches its database using perfect hashing to see whether it is present in any of the PSs in that LPSG. If the V_i is present in any of the PSs in that LPSG, then the VDM checks whether the PS in which the V_i found is neighbor to the requested PS_q [NBR(PS_q)].

If so, the VDM intimates the same to SM_{TR} which initiates the streaming of the $(\text{pref-1})_{Vi}$ from that NBR(PS_q), and $(\text{pref-2})_{Vi}$ from its cache, to the requested PS_q and the same is intimated to the requested PS_q. Then the IM_{TR} coordinate with MMS to download the remaining portion $(S\text{-(pref-1)-(pref-2)})_{Vi}$, and hence the transmission cost and the initial startup latency is very small

Otherwise, if it is not [NBR(PS_q)] but it is present in more than one PS of LPSG then SM_{TR} selects one PS such that, the path from the selected PS to PS_q should be optimum. Then it initiates the streaming of the $(\text{pref-1})_{Vi}$ from the selected PS, and $(\text{pref-2})_{Vi}$ from its cache, to the requested PS_q through the optimal path found by the SM_{TR} and the same is intimated to the requested PS_q. Thus the total transmission cost and the initial startup latency is relatively higher, but acceptable with high QoS.

If the V_i is not present in any of the PSs in that LPSG, then the IM_{TR} Passes the request to the tracker of NBR(L_p). Then the VDM(NBR(L_p)) checks its database using perfect hashing, to see whether the V_i is present in any of the PSs of its L_p. If it is present in one or more PSs, then the SM(NBR(L_p)) selects the optimal streaming path from the selected PS(NBR(L_p)) to the requested PS_q and intimates the same to IM(L_p). Then the SM(L_p) in turn initiates the streaming of V_i to the requested PS_q through the optimal path, and the same is intimated to the requested PS_q and hence the total transmission cost and the initial startup latency is comparatively high but acceptable because it avoids accessing remote MMS, which increases the network usage.

If the V_i is not present in any of the PSs of its NBR(L_p) also, then the TR(L_p) modules decides to download the V_i from MMS to PS_q. The IM_{TR} coordinates with MMS to download the remaining portion $(S\text{-(pref-1)-(pref-2)})_{Vi}$, and hence the total transmission cost and the initial startup latency very high, but very few number of videos are downloaded from MMS as shown by our simulation results.

Functionality of this load sharing algorithm is shown in Fig. 4. Whenever the sufficient buffer and bandwidth is not available in the above operation the user request is rejected, which is a very rare possibility as shown by our simulation results.

VI. Simulation Model

Our simulation model consists of a single MMS and a group of 6 TRs. All these TRs are interconnected among themselves in a ring fashion. Each of these TR is in turn connected to a set of 6 PSs. These PSs are again interconnected among themselves in a ring fashion. We use the video hit ratio(VHR), the average transmission cost to measure the performance of our proposed approach more correctly. In addition we also use, number of access to main server as performance metrics.

We assume that the request distribution of the videos follows a zipf-like distribution, the user request rate at each PS is 25 requests per minutes. The ratio of cache sizes at different elements like MMS, TR and PS is set to $C_{MS} : C_{TR} : C_{PS}$ = 10 :4:1, and transmission delay between the proxy and the client as 100ms, transmission delay between proxy to proxy as 200ms, transmission delay between TR and PS as also 200ms, transmission delay between the main server and the proxy as 1200ms, transmission delay between tracker to tracker 300ms, the size of the cached video as 280MB to 1120MB(25min – 1hr) in proportion to its popularity.

VII. Simulation Results

The simulation results presented below are an average of several simulations conducted on the model. Consider Fig.8, which shows the total number of requests arrived for videos at PSq, and the number of videos served. That is almost 80%-

82% of the served videos for these requests are streamed from LPSG through PS_q, by sharing the videos among the PSs of LPSG and NBR[LPSG]. Only about 18%-20% of the videos of these requests are served from MMS which is very less and hence the network transmission cost of the system is also reduced.

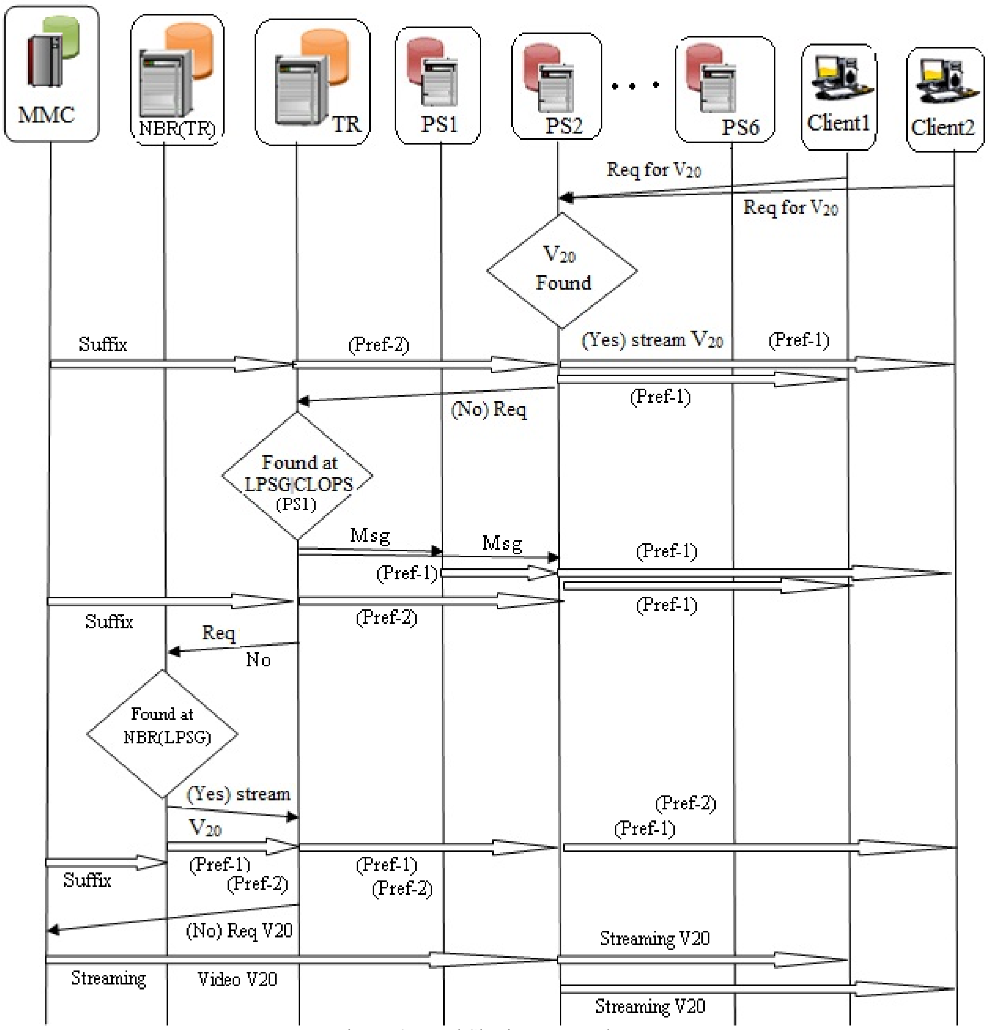

Figure 4: Load Sharing Protocol

As the (pref-1) (pref-2) of most frequently asked videos have been cached and streamed from the PS_q and (L_p+NBR[L_p]) with the cooperation of various modules of PSs, and the coordination of modules of TR of LPSG, Our scheme has achieved very high video hit ratio as shown in Fig. 5. Hence the distribution strategy of storing the videos at any one of the proxy server of LPSG only once has increased the Video availability at LPSG significantly. This in turn has increased the video hit ratio.

The proposed video prefix distribution scheme caches maximum portion (%) of more number of popular videos at LPSG close to client. These videos are streamed from LPSG itself, which in turn has reduced the number of access to remote main server, average network usage and traffic on main server to proxy path as shown in Fig. 6. Due to this reduction in network usage, transmission cost of the system is also reduced significantly when compared to the single proxy based system as shown in Fig. 7.

VIII. Conclusion

In this paper we have proposed an efficient video sharing mechanism and an architecture where proxies cooperate with each other to achieve reduced transmission cost and initial start up delay, by caching and streaming maximum portion of the most frequently requested videos among the proxies of L_p. Our simulation results demonstrated that our proposed approach has reduced the average network transmission cost of the system, initial startup delay for the videos requested at PS_q, and also the load of CMS by caching maximum portion of the most popular videos in proportion to their popularity across the PSs of L_p. And sharing of these videos among the proxies of the system also reduces the server-to-client transmission cost and time, maintains high QoS for the users and has removed the video redundancy among the proxy servers. The future work is being carried out to improve the performance using dynamic buffer management at PS_q.

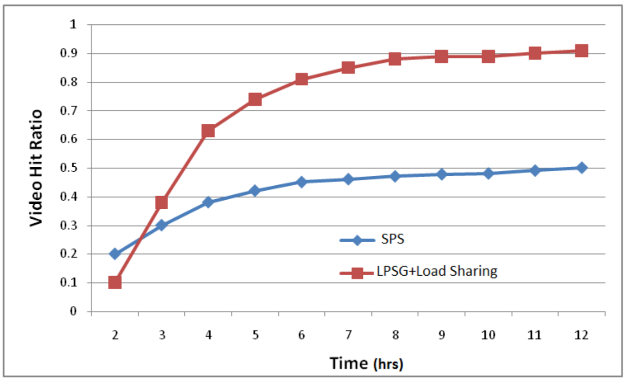

Figure 5: Video Hit Ratio Vs Time

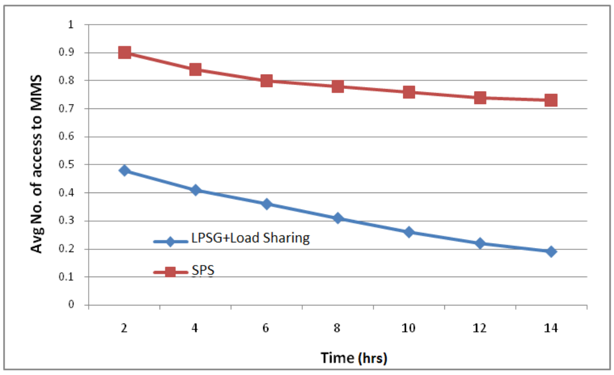

Figure 6:Avg. No. of Access to Multimedia Server

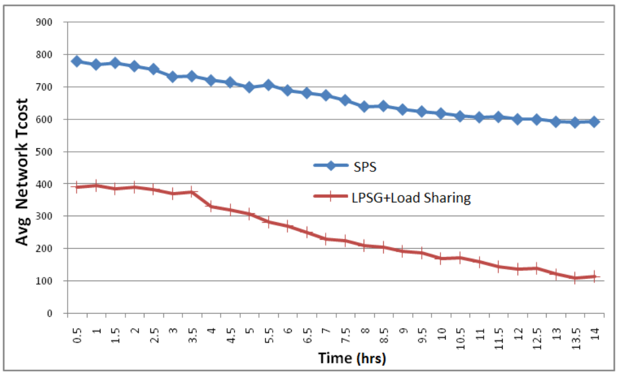

Figure 7: Average Network Transmission Cost Vs Time

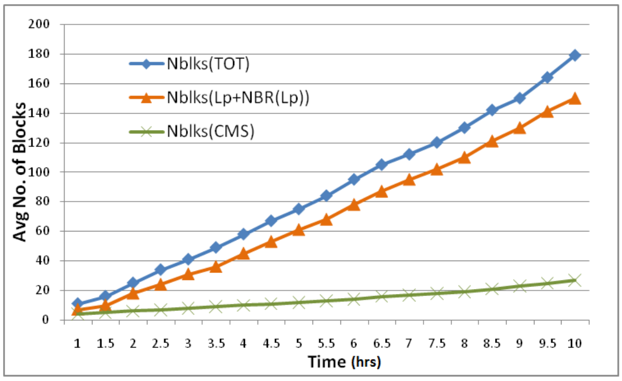

Figure 8:Avg. No. of blocks streamed from LPSG, MMS Vs Time

REFERENCES

[1] Bing Wang, Subhabrata Sen, Micah Adler and don Towsley "Optmal Proxy cache Allocation for Efficient Streaming Media Distribution" IEEE TRANSACTION ON MULTIMEDIA, VOL. 6, NO. 2, APRIL 2004.

[2] H S Guruprasad, M Dakshayini , H D Maheshappa & P Geethavani, "Video caching brother network(VCBN)for VoD Based On Popularity", *1st International Conference on Information Technology*, March 19-21, 2007, Haldia institute of Technology, Haldia,West Bengal, India.

[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] COPACC: An Architectureof cooperative proxy-client caching System for On-Demand Media Streaming. A technical report by Alan T.S lp, Jiangchuan Liu, John C.S.Lui

[7] S.-H. Gary Chan, Fouad Tobagi, "distributed Servers Architecture for Networked Video Services", IEEE TRANSACTIONS ON NETWOKING, VOL. 9,NO. 2, APRIL 2001.

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

[9] A. Feldmann, R. Caceres, "Performance of Web Proxy Caching in Heterogeneous Bandwidth Environments", In Proc. Of IEEE INFOCOM '99, March 1999.

[10] P. A. Chou and Z. Miao. Rate-distortion optimized streaming of packetized media. Technical Report MSR-TR-2001-35, Microsoft Research Center, February 2001.

[11] Dr. Mahmood Ashraf Khan, Prf.G0-Hasegawa, Yoshiaki Taniguchi "QoS Multimedia Network Architecture" murata Laboratory, Osaka University, Japan.

M Dakshayini. holds M.Tech (VTU Belgaum) in computer science, perusing Phd. She has one and a half decades experience in teaching field. She has published many papers. Currently she is working as a teaching faculty in the department of Information science and engineering at BMS College Of Engineering, Bangalore, India.

T R Gopalakrishnan Nair holds M.Tech. (IISc, Bangalore) and Ph.D. degree in Computer Science. He has 3 decades experience in Computer Science and Engineering through research, industry and education. He has published several papers and holds patents in multi domains. He won the PARAM Award for technology innovation. Currently he is the Director of Research and Industry in Dayananda Sagar Institutions, Bangalore, India.