IACSIT International Journal of Engineering and Technology Vol.1,No.5,December,2009
ISSN: 1793-8236

Efficient Resource Allocation in a Distributed VoD Using Agent Technology

H S Guruprasad and Dr. H D Maheshappa

***Abstract*—This paper proposes an efficient bandwidth allocation algorithm in which higher priority is given to the videos with higher weights using agent technology. The popularity and weight profile of the videos which is used for efficiently allocating bandwidth is periodically updated by a mobile agent. The proposed approach allocates more bandwidth for higher weight videos [popular videos], reduces the load on the central multimedia server and maximizes the channel utilization between the neighboring proxy servers and the central multimedia server and lowers video rejection ratio. The simulation results prove the reduction of load on central multimedia server by load sharing among the neighboring proxy servers, maximum bandwidth utilization, and more bandwidth allocation for higher weight videos.**

***Index Terms*—Agent Technology, Resource Allocation, VoD, Bandwidth, Multimedia**

I. Introduction

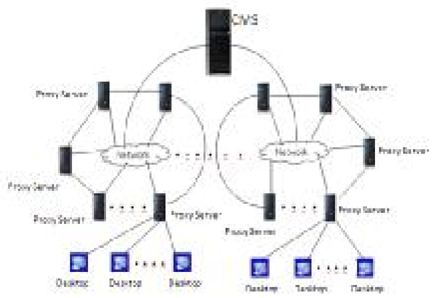
Agents are autonomous programs which can understand an environment, take actions depending upon the current status of the environment using its knowledge base and also learn so as to act in the future. Autonomy, reactive, proactive and temporally continuous are mandatory properties of an agent. The other 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, 2, 10, 12].

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 [3].

In [7], 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 [8], 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. Proxy servers are widely used in multimedia networks to reduce the load on the central server and to serve the client requests faster.

In, [2], we had considered an architecture without PSG and a comparison was made with an architecture without neighbouring proxy servers. In this paper, we propose an efficient bandwidth allocation algorithm and VoD architecture for distributed VoD system which allocates higher bandwidth to the videos which have higher weights. The 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 [PSG]. All the PSG'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.

II. Proposed Architecture



A Proxy Server Group is a group of Proxy Servers which

H S Guruprasad, Research Scholar, Dr. MGR University, Asst Prof & HOD, Dept of ISE, BMSCE, Bangalore, India. Ph: +91-9845516780
Dr. H D Maheshappa, Director, East Point Group of Institutions, Bangalore, India. Ph: +91-9916862999

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

The CMS contains all the N number of videos. These N videos are categorized into most popular, secondary popular and least popular. Initially, few most popular, secondary popular and least popular videos are loaded to the proxy servers. Also there weights for these videos are appropriately assigned.

Consider n videos $v_1,.v_2,.......v_n$. The mean arrival rates for the videos are $l_1, l_2,..., l_n$ respectively. There are m server channels. The total arrival rate of all the videos is $l = \sum_{i=1}^{n} l_i$. The probability of receiving a user request for a video v_i is given by $P_i = l_i / l$ for $i = 1,2,....,n$.

There are 3 classes of customer's c_1, c_2 and c_3 and the profit associated with each class is p_1, p_2 and p_3 respectively. Let $k_1, k_2, \mathbf{L}, k_n$ be the number of requests for the n videos $v_1,.v_2,.......v_n$. Also, $k_i = k_{i1} + k_{i2} + k_{i3}$, where k_{i1} is the number of requests of class 1, k_{i2} is the number of requests of class 2 and k_{i3} is the number of requests of class 3. Now, the weight associated with each video is $w_i = k_{i1} * p_1 + k_{i2} * p_2 + k_{i3} * p_3$.

The CMS periodically invokes a mobile agent which travels across the Proxy Servers and updates the video popularity and weight profile at the Proxy servers and the CMS.

When a request for a video arrives at the proxy server [PS], one of the following cases happens:

- The requested video is present in the PS
- The requested video is not present in the PS, but is present in either left neighbor Proxy Server[LPS] or Right neighbor Proxy Server[RPS], but not in PSG
- The requested video is present in both LPS and RPS, but not in the PS or PSG
- The requested video is present in LPS and PSG, but not in PS or RPS
- The requested video is present in RPS and PSG, but not in PS or LPS
- The requested video is not present in LPS and RPS, but is present in PSG
- The requested video is not present in LPS, RPS and PSG

If the requested video is present in the PS, then the real time transmission of the video starts immediately from the PS to the client. If the requested video is not present in the PS, then the weight of the requested video is calculated as explained above.

If the requested video is present only in LPS and not in RPS and PSG, then the bandwidth for the requested video between PS and LPS is allocated as follows:

If maximum bandwidth required for the requested video is available between PS and LPS, then maximum bandwidth is allocated. If not, minimum bandwidth is allocated if available between PS and LPS. If minimum bandwidth required is also not available for the requested video between PS and LPS, then we check if minimum bandwidth could be accumulated by deallocating excess allocated bandwidth than the minimum bandwidth starting from the bottom (i.e. least weighted video). This way excess bandwidth is taken starting from the videos which have lower weights. If minimum bandwidth could be accumulated, then this minimum bandwidth is allocated to the requested video.

If bandwidth could not be allocated between PS and LPS, then bandwidth allocation is done between PS and CMS as explained above. If bandwidth could not be allocated between PS and CMS also, then the requested video is rejected.

If the requested video is present only in RPS and not in LPS and PSG, then the bandwidth for the requested video between PS and LPS is allocated as explained above. If bandwidth could not be allocated between PS and LPS, then bandwidth allocation is done between PS and CMS as explained above. If bandwidth could not be allocated between PS and CMS also, then the requested video is rejected.

If the requested video is present in both LPS and RPS, but not in the PS or PSG, then we check for the free bandwidth available between PS-LPS and PS-RPS. If free bandwidth available between PS and LPS is more than the free bandwidth available between PS and RPS, then bandwidth allocation is done between PS-LPS, otherwise bandwidth allocation is done between PS-RPS. If bandwidth could not be allocated between PS-RPS and PS-LPS, then bandwidth allocation is done between PS and CMS as explained above. If bandwidth could not be allocated between PS and CMS also, then the requested video is rejected.

If the requested video is present in LPS and PSG, but not in PS or RPS, then the bandwidth for the requested video between PS and LPS is allocated as explained above. If bandwidth could not be allocated between PS and LPS, then bandwidth allocation is done between PS and PSG as explained above. If bandwidth could not be allocated between PS and PSG also, then the then bandwidth allocation is done between PS and CMS as explained above. If bandwidth could not be allocated between PS and CMS also, then the requested video is rejected.

If the requested video is present in RPS and PSG, but not in PS or LPS, then the bandwidth for the requested video between PS and RPS is allocated as explained above. If bandwidth could not be allocated between PS and RPS, then bandwidth allocation is done between PS and PSG as explained above. If bandwidth could not be allocated between PS and PSG also, then the then bandwidth allocation is done between PS and CMS as explained above. If bandwidth could not be allocated between PS and CMS also, then the requested video is rejected.

If the requested video is not present in PS, LPS and RPS, but is present in PSG, then bandwidth allocation is done between PS and PSG as explained above. If bandwidth could not be allocated between PS and PSG also, then the bandwidth allocation is done between PS and CMS as explained above. If bandwidth could not be allocated between PS and CMS also, then the requested video is rejected.

If the requested video is not present in PS, LPS, RPS and PSG, then the bandwidth allocation is done between PS and CMS as explained above. If bandwidth could not be allocated between PS and CMS also, then the requested video is rejected.

III. Proposed Algorith

NOMENCLATURE: PS: PROXY SERVER

LPS: Left neighbor proxy server
RPS: Right neighbor proxy server
PSG: proxy server group
BWAvail (x, y): Bandwidth available between x and y
MaxBW: Maximum Bandwidth
MinBW: Minimum Bandwidth

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

```
if the requested video is present in PS
start streaming the video from PS
  else
dynamic bandwidth allocation is done according to the algorithm DYNABAND
if bandwidth is allocated
  then
the video is downloaded and stored at PS and streamed to the requested client
  else
the request is rejected
Algorithm DYNABAND
 begin
if the requested video is present in LPS only
 then begin
call Bandwidth_allocation (LPS, PS)
if bandwidth is not allocated
call Bandwidth_allocation (CMS, PS)
  end
if the requested video is present in RPS only
  then begin
call Bandwidth_allocation (RPS, PS)
if bandwidth is not allocated
call Bandwidth_allocation (CMS, PS)
  end
if the requested video is present in both LPS and RPS
  then begin
if (BWAvail (LPS, PS)>BWAvail (RPS, PS))
  then begin
call Bandwidth_allocation (LPS, PS)
if bandwidth is not allocated
call Bandwidth_allocation (CMS, PS)
  end
  else begin
call Bandwidth_allocation (RPS, PS)
if bandwidth is not allocated
call Bandwidth_allocation (CMS, PS)
  end
  end
if the requested video is present in both LPS and PSG

then begin
call Bandwidth_allocation (LPS, PS)
if bandwidth is not allocated
call Bandwidth_allocation (PSG, PS)
if bandwidth is not allocated
call Bandwidth_allocation (CMS, PS)
 end

if the requested video is present in both RPS and PSG
 then begin
call Bandwidth_allocation (RPS, PS)
if bandwidth is not allocated
call Bandwidth_allocation (PSG, PS)
if bandwidth is not allocated
call Bandwidth_allocation (CMS, PS)
  end

if the requested video is present in LPS, RPS and PSG
 then begin
if (BWAvail (LPS, PS)>BWAvail (RPS, PS))
 then begin
call Bandwidth_allocation (LPS, PS)
if bandwidth is not allocated
call Bandwidth_allocation (PSG, PS)
if bandwidth is not allocated
call Bandwidth_allocation (CMS, PS)
 end

 else begin
call Bandwidth_allocation (RPS, PS)
if bandwidth is not allocated
call Bandwidth_allocation (PSG, PS)
if bandwidth is not allocated
call Bandwidth_allocation (CMS, PS)
 end

 end
if the requested video is not present in LPS, RPS and PSG
 then
call Bandwidth_allocation (CMS, PS)
```

end

Algorithm Bandwidth_allocation(X, PS)
begin
if MaxBW required for the video is available between X and PS
then
allocate MaxBW for the video
else
if MinBW required for the video is available between X and PS
then
allocate MinBW for the video
else
find out if MinBW required for the requested video could be accumulated by deallocating excess BW than the MinBW starting from the bottom (i.e. least weighted video)
if MinBW could be accumulated
then
allocate MinBW required for the requested video
els
BW is not allocated for the requested video

IV. Simulation Model

The simulation model consists of a single central multimedia server and a few proxy server groups. The PSG consist of a few proxy servers. The parameters considered for simulation are as follows:

Parameter	values
Number of proxy servers	6
Number of videos[NOV]	480
Bandwidth between PS-LPS, PS-RPS and PS-CMS	200MB
Max Bandwidth for the videos	30MB to 40MB
Min Bandwidth for the videos	4MB to 8MB
No. of most popular videos	NOV/4 =120
No. of secondary popular videos	NOV/4 = 120
No. of least popular videos	NOV/2 =240
No. of most popular videos initially loaded to PS	40
No. of secondary popular videos initially loaded to PS	40
No. of least popular videos initially loaded to PS	80

The performance parameters are load sharing among the proxy servers, more bandwidth allocation for the videos having more weights, video rejection ratio and the bandwidth utilization between PS-LPS, PS-RPS, PS-CMS.

V. Results and Discussion

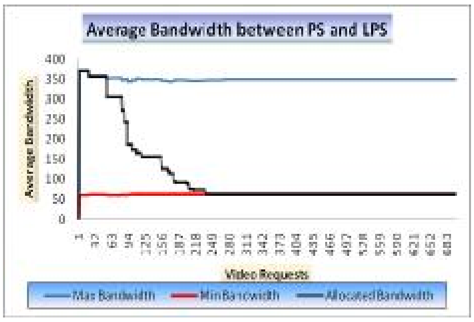


Fig 1. Average Bandwidth between PS and LPS

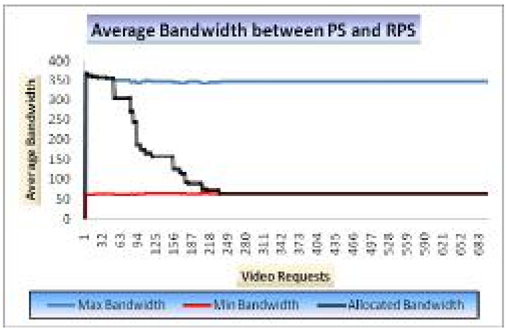


Fig 2. Average Bandwidth between PS and RPS

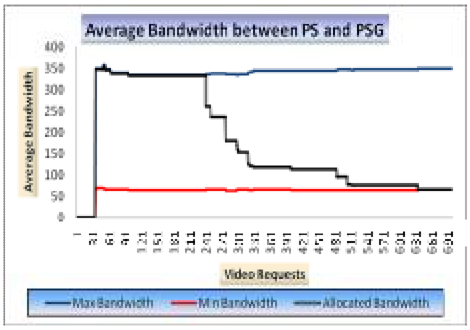


Fig 3. Average Bandwidth between PS and PSG

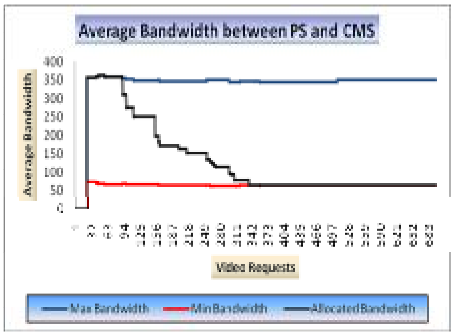


Fig 4. Average Bandwidth between PS and CMS

ISSN: 1793-8236

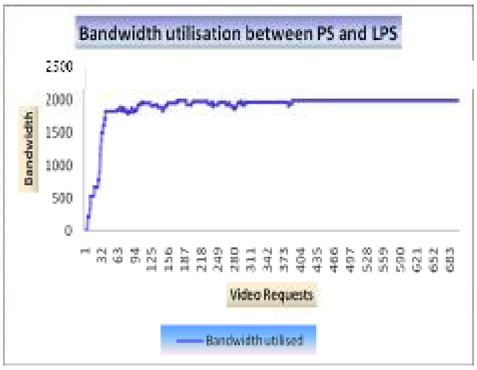


Fig 5. Bandwidth utilisation between PS and LPS

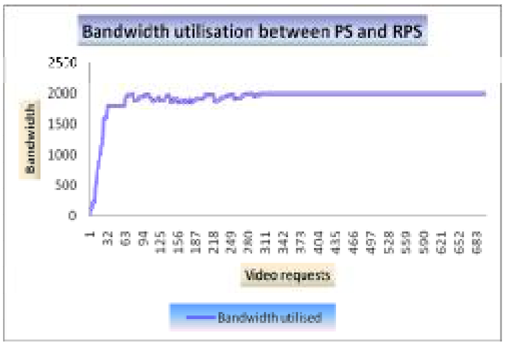


Fig 6. Bandwidth utilisation between PS and RPS

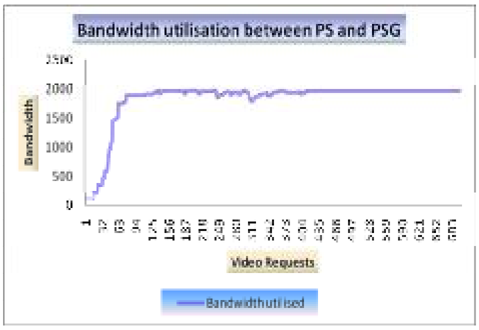


Fig 7. Bandwidth utilisation between PS and PSG

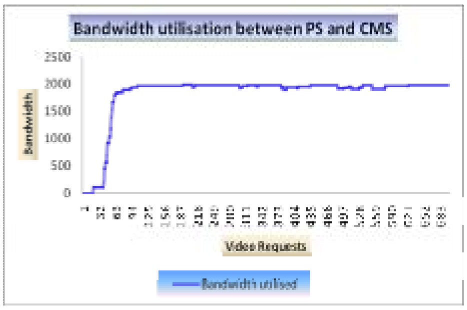


Fig 8. Bandwidth utilisation between PS and CMS

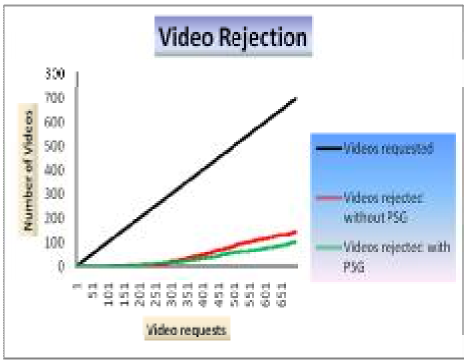


Fig 9. Video Rejection

The results presented are an average of several simulations conducted on the model. Each simulation is carried out for 10000 seconds.

Fig 1 and Fig 2 shows the average maximum bandwidth, average minimum bandwidth and the average allocated bandwidth for all the videos being downloaded from LPS and RPS respectively. Initially maximum bandwidth is allocated to the videos downloaded from LPS/RPS. Later, when the number of videos being downloaded from LPS/RPS increases, the excess bandwidth of the lower weight videos being downloaded will be taken back to allocate for new videos. Thus more bandwidth will be assigned to the more weight videos than the lesser weight videos. When the number of videos still increases, then the average bandwidth allocated still decreases.

Fig 3 shows the average maximum bandwidth, average minimum bandwidth and the average allocated bandwidth for all the videos being downloaded from PSG. Initially maximum bandwidth is allocated to the videos downloaded from PSG. Later, when the number of videos being downloaded from PSG increases[it happens whenever the requested video is not found in LPS and RPS and the video is found in PSG], the excess bandwidth of the lower weight videos being downloaded will be taken back to allocate for new videos. Thus more bandwidth will be assigned to the more weight videos than the lesser weight videos. When the number of videos still increases, then the average bandwidth allocated still decreases and becomes minimum.

Fig 4 shows the average maximum bandwidth, average minimum bandwidth and the average allocated bandwidth for all the videos being downloaded from CMS. Initially maximum bandwidth is allocated to the videos downloaded from CMS. Later, when the number of videos being downloaded from CMS increases[it happens whenever the requested video is not found in LPS,RPS and PSG], the excess bandwidth of the lower weight videos being downloaded will be taken back to allocate for new videos. Thus more bandwidth will be assigned to the more weight videos than the lesser weight videos. When the number of videos still increases, then the average bandwidth allocated still decreases and becomes minimum.

Fig 5, Fig 6,Fig 7 and Fig 8 shows the bandwidth utilisation between the PS-LPS, PS-RPS, PS-PSG and PS-CMS. The bandwidth utilisation is almost maximum as shown in the figures. Thus, the bandwidth between PS-LPS, PS-RPS, PS-PSG and PS-CMS is effiently used.

Fig 9 shows the number of videos requested, videos rejected without PSG and the videos rejected with PSG.

The number of rejections is less compared to the approach given in [2]. The number of rejections is quite low which majorly happens when the video is not found in LPS, RPS and PSG and there is no free bandwidth between PS and CMS.

VI. CONCLUSION

In this paper, we have concentrated on the load sharing among the proxy servers by considering the local proxy server group and by giving higher priority for higher weight videos using agents. The simulation shows promising results. The algorithm always uses maximum bandwidth between the neighboring proxy servers and the central multimedia server by allocating more bandwidth to the higher weight videos so that they are streamed faster. Further work is being carried out to investigate load balancing by considering cost model for VoD.

REFERENCES

[1] H S Guruprasad, H D Maheshappa, “Efficient Channel Allocation Based on the Wieghts of Videos using Agents”, JATIT [Journal of Theoretical and Applied Information Technology], Volume 5, No 4, April 2009, pp 399-405
[2] H S Guruprasad, H D Maheshappa “Dynamic Load Balancing Architecture for Distributed VoD using Agent technology” IJCSS [International Journal of Computer Science and Security], Volume (1): Issue (5), November 2008, pp 13-22.
[3] Jun Guo, Eric W.M. Wong, Sammy Chan, Peter Taylor, Moshe Zukerman, Kit-Sang Tang, “Combination Load Balancing for Video-on-Demand Systems”, IEEE transactions on circuits and systems for video technology, Vol 18, No 7, July 2008
[4] Yunpeng Chai, Lifeng Sun, Zhihui Du and Sanli Li, “Load Sharing based on PSO Algorithm for Isolated Distributed Stream Servers”.
[5] Santosh Kulkarni “Bandwidth Efficient Video on Demand Algorithm (BEVA)” 10th International conference on Telecommunications, Vol 2 pp 1335-1342, 2003
[6] 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
[7] 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.
[8] 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.
[9] King-Man Ho, Wing-Fai Poon, Kwok-Tung Lo, “Video-on-Demand Systems With Cooperative Clients in Multicast Environment”, IEEE Transactions on circuits and systems for video technology, Vol 19, No3, March 2009
[10] Mohammed A. M. Ibrahim, “Distributed Network Management with Secured Mobile Agent Support”, International Conference on Hybrid Information Technology (ICHIT'06), 2006
[11] 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
[12] S S Manvi and P Venkataram, “Mobile Agent based online Bandwidth allocation Scheme in Multimedia Communications”, IEEE GLOBECOM 2001 Conference USA
[13] A Dan, D Sitaram and P Shabuddin, “Dynamic batching policies for an on demand video server “, Multimedia Systems, pp 51-58, 1996
[14] Pushpendra Kumar Chandra, Bibhudatta Sahoo, “Performance Analysis of Load Balancing Algorithms for cluster of Video on Demand Servers”, IEEE International Advance Computing Conference (IACC 2009), Patiala, India, 6-7 March 2009.

H S Guruprasad is an Assistant Professor and Head of the Department of Information Science & Engineering, BMS College of Engineering, Bangalore, India. He is an Engineering Graduate from Mysore University and did his Post Graduation at BITS, Pilani, India in 1995. He has vast experience in teaching and has guided many post graduate students. His research interests include multimedia Communications, distributed systems, computer networks and agent technology.

Dr. H.D. Maheshappa Graduated in Electronics & Communication Engineering from University of Mysore, India in 1983. He has a Post Graduation in Industrial Electronics from University of Mysore, India in 1987. He holds a Doctoral Degree in Engineering from Indian Institute of Science, Bangalore, India, since 2001. He is specialized in Electrical contacts, Micro contacts, Signal integrity interconnects etc. His special interests in research are Bandwidth Utilization in Computer Networks. He has been teaching engineering for UG & P G for last 25 years. He served various engineering colleges as a teacher and at present he is the Director for East Point Group of Institutions, Bangalore, India. He has more than 35 research papers in various National and International Journals & Conferences. He is a member of IEEE, ISTE, CSI & ISOI. He is a member of Doctoral Committee of Coventry University UK. He has been a Reviewer of many Text Books for the publishers McGraw-Hill Education (India) Pvt., Ltd, Chaired Technical Sessions, National Conferences and also has served on the advisory and technical national conferences.