Chaining Algorithm and Protocol for Peer-to-Peer Streaming Video on Demand System

Hareesh.K[1] and Manjaiah D.H[2]

[1]Research Scholar, Jawaharlal Nehru Technological University, Anantapur, A.P, India
mail_hareeshk@yahoo.com

[2] Professor and Chairman, Department of CS, Mangalore University, Mangalore, India.
ylm321@yahoo.co.in

ABSTRACT

As the various architectures and protocol have been implemented a true VoD system has great demand in the global users. The traditional VoD system does not provide the needs and demands of the global users. The major problem in the traditional VoD system is serving of video stream among clients is duplicated and streamed to the different clients, which consumes more server bandwidth and the client uplink bandwidth is not utilized and the performance of the system degrades. Our objective in this paper is to send one server stream sufficient to serve the many clients without duplicating the server stream. Hence we have proposed a protocol and algorithm that chains the proxy servers and subscribed clients utilize client's uplink bandwidth such that the load on the server is reduced. We have also proved that less rejection ratio of the clients and better utilization of the buffer and bandwidth for the entire VoD system.

KEYWORDS

Video-on-Demand (VoD), video, Chaining, Segment, Buffer, Bandwidth.

1. INTRODUCTION

The internet is a global system, which serves the billions of global users today across worldwide. The users are access the vast range of information resources, services and new features from it. The most traditional communications such as telephone, music, film and television are redefined by the internet and provide new features and services include voice over Internet protocol (voIP) and IPTV. IPTV[1][2] is a new system provides lot of features and multimedia services likes Live TV, Video on demand(VoD) and interactive TV etc., reliably and secure in quality content distribution over IP based networks than traditional cable television fails to provide features such as high definition video quality and VCR functionalities. Another important aspect that makes IPTV so popular and better than traditional cable television is Video on Demand (VoD) system. The applications involves, streaming video files from a multimedia server to the client systems that is large number of users on an individual basis [3][4][5]. This process involves streaming of videos requested by the clients in particular to them. In regular cable TV certain kind of customized viewing option is absent as the client received the requested video. The client requests for a scrupulous video on a multi media server and once the request is received by the multimedia server, which is process the request and served to the requested client. Through VoD, the video can be viewed as it is being downloaded from the server. In real work loads, high degrees of interactive operations were

DOI : 10.5121/ijwmn.2011.3611

observed [6]. Along with this the VCR functionalities provide an extra edge over conventional television, letting the user decide when and how to watch the video.

The implementation of VoD requires a server with high bandwidth network framework which has sufficient storage capacity to store video files on the VoD network. One of the main requirements of a VoD network which uses IPTV generally uses High definition TV videos and hence servers need to have large storage capacities to store the video files. Using MPEG[7] standards the bandwidth required is 2133 Mbps. To keep up such a high bandwidth this requires a high bandwidth VoD network. This makes the service costly and not viable to implement from the client's side. To conquer these drawbacks the VoD network is implemented in three stages.

- Stage I has the media server with a fiber optic communication channel having in the range of 2 Gbps bandwidth.
- Stage II consists of cluster of proxy servers which are connected to the Media server at one side and group of clients on other side. The connection on the media server side is a fiber optic communication channel and that on the client side is a low bandwidth network that provides a bandwidth in the range of 2 Mbps.
- Stage III consists of a group of clients that are served by the proxy server and in turn served by the media server.

Initially, suppose if a client makes a request for a video to the media server then the server downloads the entire video data file to nearest proxy server of the requested client. This proxy server in turn requests to the server for scrupulous movie then server will streamed the movie data file to the proxy server and then streamed to the requested client.

When a client is requested a movie to the server that is previously being streamed to some client, again by the other client request for the same movie file thereby streaming the same movie to another clients that uses twice the bandwidth for the same movie file. The server bandwidth is of great importance in the VoD network and the streaming of the same movie twice which is not an optimal utilization of the server bandwidth. On the other hand client's side uplink bandwidth is not utilized. To overcome these draw backs we have provide a protocol for the better utilization of client's uplink bandwidth for streaming of the movies and hence reducing the load on the server which is our main objective of this paper.

2. RELATED WORK

The VOD streaming has attracted significant attention on many researchers from past decade; some of the previous works are as follows. There have been two kinds of works in this regard. They are: Building of Streaming servers Protocol and architecture related works.

Here, we are proposing an algorithm and protocol; our work will focus more on Protocol and Architecture related works. There have been quite a few innovative techniques and protocols proposed in this field. In prefix caching [8] concentrates more on the storage problems related to proxy servers, but in our work, when a client makes a request to the proxy server. It checks in the current movie list of the proxy server. If the movie is available then it streams from the proxy server to the requested client, otherwise the movie is downloaded from the media server to the proxy server and then the movie is streamed to the requested client. In the present days where memory is cheap and easily available the impact of prefix caching is very less and the proxy server which are used having more storage space.

In optimized content distributed delivery [9] method a single multimedia file is split into a several parts and transmits in parts from the various different servers to the requested clients. Effectively this is like many clients are served by the various different servers. The drawback of these systems is that, the server bandwidth is costlier and very expensive to maintain the multiple servers. In our work, the proxy server has classes of clients based on the client's resources and scarce uplink and downlink bandwidth with limited buffer. These proxy servers are connected to the main media server through T1/E1 link. Each client acts as a server and has sufficient resource to store the video stream as well as forward the buffer to the nearer clients using its uplink bandwidth. By this way, more number of clients is chained for single server stream and which is less expensive inturn reduce the server bandwidth.

In hybrid model [10] uses many clients called peers-lend or supply their resources to the network and serve these resources to their peers. This model efficiently uses all the network resources and provides very good services but the major drawback is that, the protocol requires the services among all the peers in a network and hence the service becomes not reliable and the model is too difficult to implement. In our work, the optimal utilization of the server bandwidth is proposed using protocol and implemented using Chaining algorithm. Hence the protocol requires services of each and every peer and the service become more reliable.

To further improve the proxy or server load resources, in [11] proposed a client-side cache scheme called earthworm. In this scheme, the client not only playbacks the movie but also forwards the stream to another client with adequate buffering delay. This is called as basic chaining. Further, this can be combined with batching period, which is called as extended chaining [12] in both the schemes, it uses backward buffering. Therefore, when client buffer overflows, then the chain has to be broken and a new server stream is required. To address this problem, an adaptive chaining [13] scheme is used. This achieves double the performance of the above-mentioned chaining schemes, in which both forward and backward client buffers are used for streaming the movies. However, the combined length of the forward and backward buffers of the client is less than length of the video segment. Hence, to extend the chaining as much as possible, an optimal chaining [14] is proposed in which the buffer of the remaining clients is exploited. This is achieved by patching stream during playback of the first few video segments. The major drawback of this scheme is over-utilization of the bandwidth without client request. The buffer of the neighboring clients is exploited by assigning the segments. Due to this, tremendous transfer of same data occurs in the network, which will degrade the overall performance of the network. In our architecture, we combine the non-scaled proxy server based architecture and decentralized P2P overlay network which reduces the overall cost of the system and improve the system performance.

3. SYSTEM ARCHITECTURE

The system network is the combination of proxy based architecture and peer to peer network based system as shown in Figure 1.The network architecture consists of a media Server, a cluster or region we find Proxy servers and many clients are connected through proxy server to the media server. Each Proxy server in a cluster act as a regional server which serves number of clients that is available in a particular region. One of the aspects of having proxy servers is that the probability of more than one client requesting the same movie in a given region.

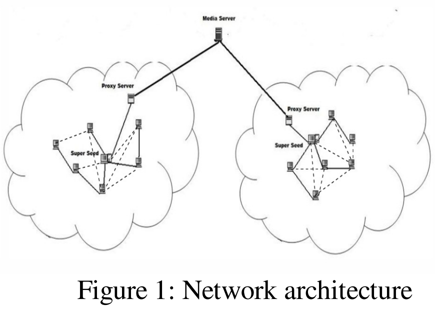

Figure 1: Network architecture

Since, if the movie is more popular, the probability of clients requesting the same movie will increase. Hence, quite obvious that number of clients requesting for popular movie will be greater in number. Thus if the Server satisfies each and every request for the same movie, the server bandwidth is not optimally utilized. In order to avoid this problem, these popular movies will be stored in the regional Proxy server buffer and streamed as and when the client requests for it. Since the popularity of each movie is depends on the region based, each proxy server buffer contains the movies which are popular in a given region and the movies can be streamed directly to the requested clients. The popularity is based on the number of hits that occur for a particular movie. The movies are stored dynamically in the proxy server based on the popularity, that if the popularity of a particular movie will decrease, the movie is automatically deleted from the proxy buffer and in turn provide to some other more popular movie. The above mentioned parameters comprise the essential requirements for the proposed protocol.

Whenever a movie is requested by a client, instead of being served by the server or the proxy server, the request is redirected to another client who already contains the same video blocks in its buffer; this process is called Chaining. This reduces the load on the proxy and the media server. Hence the performance of the system increases and better utilization of the server uplink bandwidth is implemented using Chaining algorithm.

3.1 Proposed Protocol

In this section describes how the protocol and algorithm is implemented. Whenever a client requests for a new Movie Id to the request handler, it checks the user details, and passes that request to media server. Media server checks existence of requested movie in its buffer. If the requested movie exists then the media server communicates with chaining handler to precede the serving operation. In the absence of movie in its buffer, chaining handler initiates proxy navigation module or cache module in order to get requested movie. The chaining handler checks active and passive list of clients and intimates client details that stores same movie (within its region) to requested client, and thereby initiates chaining operation. In case of absence of requested movie in both active and passive list of clients, proxy server starts streaming operation of requested movie to requesting client.

Proxy server executes given algorithm, upon receiving a movie request from a client. First proxy checks requested movie in its active list of clients. Then proxy server checks for requested movie in its passive list of clients In case of absence of requested movie file, proxy server initiates searching operation in its nearest proxies The proxy server executes pop check procedure to check popularity of given movie falls below the threshold values. Proxy server compares popularity, timestamp values to find out the existence of given movie in client video buffer. If popularity of a movie increases, automatically the life time of that movie in client video buffer also increases. In the absence of movie in client video buffer, proxy server starts streaming operation of movie. A client system, after receiving a chaining request from proxy server, executes given client chaining algorithm, as specified below.

3.2 Algorithm

The algorithms that are required to implement the above protocol are as follows:

CHAINING ALGORITHM

Nomenclature:
C= $\{C_1, C_2, C_3, C_4 \ldots\ldots C_m\}$, Active clients **P**= $\{P_1, P_2, P_3, P_4 \ldots\ldots\ldots P_n\}$, Active Proxies
C_i →Client address
P_i →Proxy address
B_i →Buffer size used in client C_i

$b= \{(\mu_1, \vartheta_1), (\mu_2, \vartheta_2)\ldots (\mu_i, \vartheta_i)\}$, b is a list containing μ_i and ϑ_i
Where μ_i is Movie Id, ϑ_i is popularity.
D {i, j} →Distance between Client C_i and C_j
μ→ Movie Id (Request Id)
η_i→ Uplink of client C_i
R = { r_1, r_2, r_3……r_k}, A list containing current streaming request.
Bi→ Proxy P_i's current Bandwidth
β→Maximum allowable buffer size of proxy P S_r→ Requested segment
Si→ Segment number i
Q→A circular queue in client holding received segments

Proxy Server

```
Upon receiving request from client $C_i$
if $\mu$ in R then
find j such that (cost*i+ → D {i, j}/ $\eta_i$) < D {i, p} / $B_i$, where $r_j$→$\mu$
if ∃ some j such that $C_j$ ϵ C
        chain client $C_i$ to $C_j$
        $\vartheta_i$ → $\vartheta_i$+1, where ri→ $\mu$
else
  if $\mu$ in b then
        initiate new stream $r_i$ to client $C_i$,
        R=R U {$r_i$}
        $\vartheta_i$ → $\vartheta_i$+1, where $r_i$→ $\mu$
else
   send request $r_k$→ $\mu$ to server
   receive segments from server
if maxsize{b} > $\beta$
    replace ($\mu_k$, $\vartheta_k$) in b with ($\mu_l$, $\vartheta_l$)
    Where $\vartheta_k$< $\vartheta_j$ ∀ j
```

Client

```
Send a request $r_i$ to proxy $P_j$
Upon receiving SIG_READY from $P_j$
Receive segment So from $P_i$,Q= Q U {So}
cptr=0,i=0
if $B_i$ < size($S_i$)
        Q= Q - {$S_{in}$}
        Q= Q U {$S_i$}
        cptr →cptr+1
        Upon receiving SIG_CHAINTO
If $S_r$ in Q
        stream the segments $S_r$ to $S_n$
else
reject
```

If both the clients belong to same region, and requested movie exists in client video buffer then chaining operation starts. If both the clients belong to different region, chaining not possible message is given to proxy server. In the absence of requested movie file in client video buffer, chaining request is rejected. If the movie is not available in the proxy server or not available in any of the serving client, then the initial portion of the video blocks are directly streamed from the main server and the later portion of the video blocks are first downloaded and buffered in the proxy server and then streamed from the proxy server to the requested client. Perceptive behind this idea is all the customers and services providers are profited by better utilization uplink bandwidth and the system resources.

4. SIMULATION AND RESULTS

The following are the assumptions made in the simulation model: It consists of a single server, 5 proxy servers and 80 clients. The server had 20 movies each of size 4.9 MB. The proxy buffer was large enough to hold 50% of the data on the server that is effectively10 movies. The server to proxy bandwidth is 2 Gbps and the proxy to client bandwidth is 2Mbps. The uplink bandwidth of the client is 256kbps.

The graph shown in Figure 3 is the combined graph of "Number of clients' v/s Time" and the "Number of streams v/s Time".

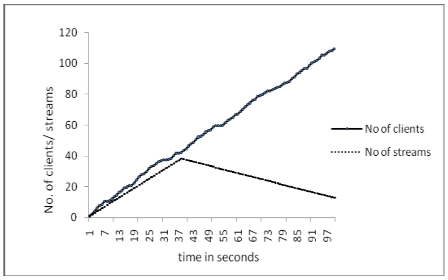

Figure 3: Number of clients and Number of Streams v/s time

We can observe that as the time increases the number of clients will also increase. But as the number clients increases we can observe that the number of streams increases initially and then decreases gradually. This is because the proxy or the server serves the clients only till all the popular movies have been cached in at least one place. Later on the movies are chained from these places.

The graph in Figure 4 shows the number of requests /streams with respect to time. In this case the requested movie is very popular and hence with time the number of requests for the movie is increased. We can observe that as the number of requests increases, the number of active chains also increases.

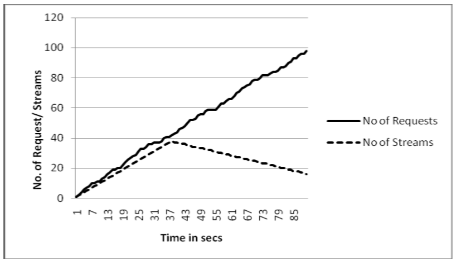

Figure 4: Number of streams v/s time

Initially the movie is present in very few clients, who are scattered in different locations. Hence the more number of active chains have to be created in order to serve the requests. But as the requests are served, the number of clients that have the movie will grow. Once more clients possess the movie, the number of active chains gradually decreases. The same as with the streams allocated for the movie in the server as shown in the graph. But the number of streams is always less than the number of active chains (as seen in the graph, the number of streams is always below the number of active chains) because of the chaining. The chaining reduces the number of streams from the server and hence the server bandwidth is conserved.

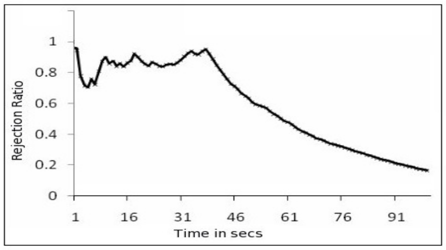

Figure 5: Rejection ratio with time

The graph in Figure 5 shows Rejection Ratio with respect to time. The rejection ratio is defined as the ratio of the number of rejected requests to the number of active chains. It represents the fact that the client requests are rejected even though there is a possibility of chaining. This happens because even though the chaining is possible the cost of chaining exceeds the actual cost of streaming. The ratio is around 0.9 initially since the number of clients possessing the movie is small and scattered. But as the number of clients that possess the movie increase the rejection ratio comes rapidly as shown in the graph. We can observe that the bandwidth initially increases with time and then peaks at a point and then reduces gradually to some constant value.

Initially the numbers of requests that arrive have to be served by the server itself because the number of clients that possess the movie is small, as shown in the previous graph of the rejection ratio.

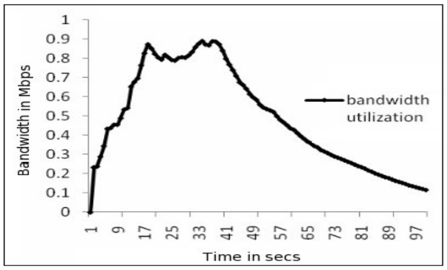

Figure 6: Bandwidth utilization with time

But once the chaining process is activated the bandwidth used of the server is reduced and hence the burden on the server is reduced to a great extent.

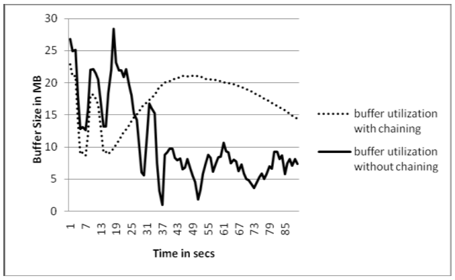

Figure 7: Buffer utilization

The buffer utilization graph shown in Figure 7 shows the client buffer utilized. Without chaining we can observe that the buffer utilization is erratic and decreases drastically with time. But with chaining we can observe that the buffer utilization is initially high and then reaches the peak and then falls down gradually. The peak represents the peak of the chaining process. After the completion of the chaining and the after the movie has been fully watched by the client the buffer utilization falls; this is the dip in the graph.

5. CONCLUSION

The Chaining reduces the overall system bandwidth and buffer utilization. In this paper we studied the existing architecture such as content delivery network; proxy based approach and peer to peer overlay networks, which suffers from the low performance by over utilization of bandwidth and buffer for the same movie. To improve the performance, we studied basic, extended, adaptive and optimal chaining schemes. The drawback of these schemes is the utilization of bandwidth is not efficient hence in this paper we have proposed a protocol and algorithm for better streaming of video streams through chaining process to the requested client using the client uplink bandwidth. Through simulation we have proved the advantages of using this protocol in the real world environment. The protocol uses the clients' uplink bandwidth, the server bandwidth is optimized and also the client's uplink bandwidth that was previously under used is well utilized.

We intend to build on the current system and make some improvements to make the rejection ratio even lesser as future enhancements. This will include procedures that will consider the popularity and the cost of streaming, in order to dynamically maintain the movies in the proxy and the servers. The algorithms used in this protocol are not equipped to handle fault tolerance. In case the connection between any two clients, which are involved in a chain, breaks, our algorithm does not provide a mechanism to overcome the drawback.

REFERENCES

[1] B. Alfonsi." *I want my IPTV: Internet Protocol television* "predicted a winner, 2005.

[2] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatrianin. "*Watching Television Over an IP Network". * In Proceedings of ACM IMC, 2008.

[3] J. Liu, B. Li, and Y.-Q. Zhang, "*Adaptive video multicast over the Internet*," IEEE Multimedia, vol. 10, no. 1, pp. 22-33, Jan./Feb. 2003.

[4] D.-E. Meddour, M. Mushtaq, and T. Ahmed. "*Open Issues in P2P Multimedia Streaming*".MULTICOMM2006 Proceedings, pp.43-48, 2006.

[5] X. Zhang, J. Liu, B. Li, and T.-S. Yum, "*CoolStreaming/DONet: A data-driven Overlay Network for Live Media Streaming*". IEEE INFOCOM'05, Miami, FL, USA, pp.2102-2011,March, 2005.

[6] RochaM,MaiaM, Cunha I, Almeida J, Campos S "*Scalable media streaming to interactive users*". Proc. ACM Multimedia, pp 966–975, November 2005

[7] ISOIIEC JTCI/SC29 WG111602, Information Technology –"*Generic Coding of MovingPictures and Associated Audio*", Recommendation H.262, ISOIIEC 13818-2 CommitteeDraft, November 4, 1993.

[8] Bing Wang, Subhabrata Sen, Micah Adler, and Don Towsley, "*Optimal Proxy Cache Allocation for Efficient Streaming Media Distribution*", IEEE Transactions on Multimedia, Vol. 6, No. 2, April2004.

[9] Gerassimos Barlas and Bharadwaj Veeravalli, "*Optimized Distributed Delivery of Continuous-Media Documents over Unreliable Communication Links*", IEEE Transactions on Parallel and Distributed Systems, Vol. 16, No. 10, October 2005

[10] Mohamed Hafeeda and Bharat Bhargava," *A Hybrid Architecture for Cost Effective On-Demand Media Streaming*", FTDCS 2003.

[11] Hua, K.A., Sheu, S. and Wang, J.Z. '*Earthworm: a network memory management techniquefor large-scale distributed multimedia applications*', Proc. IEEE INFOCOM'97, Vol. 3, Kobe,Japan, April, pp.990–997

[12] Sheu, S., Hua, K.A. and Tavanapong, W. (1997) '*Chaining: a generalized batching technique for video-on-demand systems'*, Proc. IEEE Int. Conf. Multimedia Computing and Systems,Ottawa, ON, Canada, pp.110–117.

[13] Chen, J.K. and Wu, J.L.C. (1999) '*Adaptive chaining scheme for distributed VOD applications*',IEEE Trans. Broadcast., Vol. 45, No. 2, pp.215–224.

[14] Te-Chou Su, Shih-Yu Huang, Chen-Lung Chan, and Jia-Shung Wang,"*Optimal Chaining Scheme for Video-on-Demand Applications on Collaborative Networks*", IEEE Transactions on Multimedia, Vol. 7, No. 5, October 2005.

[15] K. A. Hua, S. Sheu, and J. Z. Wang Earthworm: "*A network memory management technique for large-scale distributed multimedia applications*", in Proc. IEEE Infocom'97, vol.3, Kobe, Japan, Apr. 1997.

[16] J. K. Chen and J. L. C. Wu, "*Adaptive chaining scheme for distributed VOD applications"*, IEEE Trans. Broadcast., vol. 45, no.2, pp. 215–224, Jun. 1999.

[17] Te-Chou Su, Shih-Yu Huang, Chen Lung Chan, and Jia-Shung Wang, "*OptimalChaining Schemefor Video-on-Demand Applications on Collaborative Networks*" ,IEEE Trans. multimedia, vol. 7, no. 5, october 2005.

[18] Hyunjoo Kim and Heon Y. Yeom*," P-chaining: a practical VoD service scheme autonomic ally handling interactive operations, Multimedia Tools Appl"*, Springer Science, Business Media, LLC2007.

[19] Frederic Thouin and Mark Coates, McGill University, "*Video- on- Demand Networks: Design Approaches and Future Challenges",* IEEE Network, March/April 2007.

[20] Schultz JJ, Znati T, "*An efficient scheme for chaining with client-centric buffer reservation for multi-media streaming",* Proc. of the 36th Annual Simulation Symposium (ANSS'03), pp 31–38,2003.

Authors

Short Biography

Hareesh K[1] is a Research Scholar, Jawaharlal Nehru Technological University, Anantapur, A.P, India. He received his BE in Computer Science and Engineering from Bangalore University and MTech in Computer Science and Engineering from Visvesvaraya Technological University, Belgaum, India. His area of interest includes Computer communication network, web Technology, mobile ad hoc networks, multimedia applications, database management systems, Internet technologies and quality of service. His current research includes P2P Video-on-Demand systems.

Manjaiah D.H.[2] is a Professor and Chairman in the Department of Computer Sciences, Mangalore University, Mangalore, India. He received his Ph.D degree from University of Mangalore, M.Tech from NITK, Surathkal and B.E. from Mysore University, India. He has more than 15 years extensive academic, Industry and Research experience. He has authored more than 50 research papers in International / National reputed journals and conferences. His current research focuses on protocol validation and verification, network performance and analysis, Advanced Computer Networking, Mobile / Wireless Communication, Wireless Sensor Networks, Operations Research, E-commerce, Internet Technology and Web Programming.