ADAPTIVE BATCHING SCHEME FOR MULTICAST NEAR VIDEO-ON-DEMAND (NVOD) SYSTEM

*Madhu Jain , **Vidushi Sharma and **Kriti Priya

*Department of Mathematics, Institute of Basic Science
Dr. B.R. Ambedkar University, Agra-282002, India

**Institute of Technology and Science,
Mohan Nagar, Ghaziabad-201005, India

priyakriti@yahoo.com

ABSTRACT

Video-on-Demand is becoming a most sought after multimedia application. It is difficult to attain a true video-on-demand (TVOD) system, so near video-on-demand (NVOD) is catching everyone's attention. In NVOD, requests are multicast in different streams. Important issue in this system is the choice of batching time. Traditionally, the batching time is fixed depending on the number of requests. In this paper we have suggested an adaptive batching scheme (ABS) where batching time is adjusted according to the current arrival rate, which follows the hyper-exponential distribution pattern. A comparison is made between the fixed and adaptive batching schemes. Numerical illustrations are provided to show that adaptive batching policy is better than fixed batching policy for optimizing bandwidth requirements.

Keywords: Near Video-on-Demand (NVOD), Bandwidth requirement, Adaptive batching, Multicasting streams, Hyper-exponential input.

1.0 INTRODUCTION

Recent advances in computing and computer networks have made real-time multimedia application delivery possible. Video-on-Demand (VOD) is one such application which has come up due to the advent of telecommunication systems. VOD is a multimedia system which provides interactivity between end users and a multimedia server. The user can select a movie from the video database and can access it, utilizing the control functions like fast forward, rewind, pause, etc. This type of system is a true Video-on-Demand (TVOD) system. Designing a TVOD system is a complicated task, as it requires a very large bandwidth to serve a large number of users. It is not feasible to provide a dedicated transmission line to each of the users because it would incur a huge sum of money. Due to these cost constraints, near Video-on-Demand (NVOD) is becoming popular, as it is comparatively less expensive. In order to develop a cost effective system a NVOD is developed. In NVOD, the users that request the same movie are put in a particular batch and each batch will be served using multicasting. Dan, Sitaram, and Shahabudin (1994) proposed a client pull scheduling algorithm called batching. Batching helps the server to serve more requests at the same time thereby increasing the capacity of service. The multicast stream used to serve the customers of a batch is opened at a particular time called the batching time. Batching time has to be fixed so that more and more users can be served with the multicast stream. On the other hand, batching introduces extra delay to the early requests where they are being held longer at the scheduler buffer, waiting to be processed. In case a customer (user) arrives after the opening of the multicast stream, he is served via a dedicated unicast stream with a faster transmission rate. As soon as the unicast stream comes in synchronization with the multicast stream, the user is merged in the previous multicast stream. The selection of the batching time greatly affects the performance of such system. If the arrival rate of the customers is small, and the batching time is also small then the bandwidth is not optimally utilized; on the contrary if the traffic or arrival rate of customers is large and the batching time is large, the performance of the system degrades. Also, if the batching time is kept very small, under heavy traffic conditions, a large number of multicast streams will be formed and if the batching time is kept very large in heavy traffic conditions, then to serve new customers, more number of unicast streams will be formed, and again the bandwidth capacity of the network will be depleted.

In real time VOD system, the demand for movies by customers may occur in phases, so that input process will be considered to follow phase type distribution. In this paper, we have proposed a hyper-exponential distribution pattern for the arrival rate of customers. The related work is discussed in section 2. In section 3, an analytical model for fixed batching scheme for obtaining the average number of streams required for the system is given. Adaptive batching pseudocode is proposed in section 4. Numerical illustrations are given in section

5 in order to facilitate comparison of the fixed and adaptive batching scheme. Finally the conclusion is drawn in section 6.

2.0 RELATED WORK

In recent years a lot of work has been done in this area. Wong, Zhang, and Pang (1995) discussed VOD service policies. Gemmel, Vin, Kandlur, and Rangam (1995) studied and surveyed multimedia storage servers during high traffic conditions. Aggrawal, Wolf, and Yu (1996) gave optimal policies for VOD storage servers. They further extended their work on adaptive piggybacking which is a technique for data sharing in a VOD storage server. In this technique various copies of videos are placed on the server with different playout rates. These video copies are transmitted in such a manner so that it can merge with the customers together in a batch. Scheduling techniques on a VOD server was studied by Sheu, Hua, and Hu (1997). Since it is difficult to attain a TVOD system, a novel video layout strategy for NVOD systems was given by Chen and Thapar (1997). A lot of work focuses on channel allocation problems in VOD. Kim and Zhu (1998) discussed these problems using adaptive piggybacking. Multicasting techniques help in providing services to more customers at a particular time. Hua, Cai, and Shen (1998) gave multicasting techniques for VOD services. Further instantaneous VOD services using controlled multicast was given by Gao and Towsley (1999). In this scheme there was restriction on the arrival time of the video. Only those requests that arrive within time interval 'T' from the start time of multicast stream can join the current multicast channel for video. Otherwise a new unicast channel is intialised to deliver the whole video stream to the client. Poon and Lo (1999) discussed batching policies in multicast systems. They proposed the concept of double rate batching policy in which customers who request the same video or audio file, are grouped together in batches and the file is transmitted using multicasting via a single transmission stream. The multicast streams are opened after a particular time, which is called as the batching time. In case a customer (user) arrives after the opening of the multicast stream, he is served via a dedicated unicast stream with a double transmission rate. As soon as the unicast stream comes in synchronization with the multicast stream, the user is merged into the previous multicast stream.

Poon, Lo, and Feng (2001) gave an adaptive batching scheme for multicast VOD systems. They gave an algorithm to find out the optimal batching time by the newly updated arrival rate of customers so as to minimize bandwidth requirement. They assumed the arrival rate of customers to be Poisson distributed. Generally, for modeling internet traffic, Poisson distribution is used but the high variability of the traffic in VOD systems cannot be achieved by the Poisson distribution. For description of Internet traffic, other distributions have also been proposed like Pareto, Weibull and etc. (Feldmann & Whitt, 1998). To characterize the variability (burstiness) in real-time communication systems, Porhuri, Petr, and Khan (2002) suggested the general hyper-exponential inter-arrival distribution for modeling network traffic. Recently Jain and Kriti (2003) gave a batching strategy for multicast stream in VOD systems by assuming the arrival rate of the customers to be a non-linear function of time.

3.0 FIXED BATCHING SCHEME

The analytical model for calculating the average number of streams required in the double-rate batching scheme is developed by assuming the hyper-exponential distribution pattern for input. The following notations are used to formulate the batching scheme:

λ_i Mean arrival rate for ith customer
L Length of the movie
W Batching time (in minutes)
C Bandwidth for a transmission line or stream (bits/minute)
β Average number of customers arriving within batching time W
B Bandwidth requirement (in bits/minutes) for one multicast group for the whole movie
N Number of streams required for double-rate batching scheme

To compute the bandwidth requirement (B), the following assumptions are taken into consideration:

- The first b customers are the unicast customers who require 2C bandwidth (each) in order to double the transmission rate.
- The customers in the multicast stream need C bandwidth.

Since we assume the arrival process as hyper-exponential with n stages having arrival rates as $\lambda_1, \lambda_2, ..., \lambda_n$ respectively, β can be calculated as

$$\beta = \max(\lfloor \lambda * W \rfloor, 1) \qquad ...(1)$$

where $\lambda = \lambda_1 + \lambda_2 + ... + \lambda_n$ and $\lfloor x \rfloor$ denotes the greatest integer less than x.

Therefore, B can be calculated as

$$\begin{aligned} B &= x(2C) + (2x)(2C) + \dots + (\beta x)(2C) + (L\text{-}2x)(C) \\ &= 2Cx(1 + 2 + \dots + \beta) + (L\text{-}2x)(C) \\ &= Cx(\beta)(\beta+1) + (L\text{-}2x)(C) \end{aligned} \quad \dots(2)$$

It should be noted that in the last term, C is multiplied by L-$2x$ because the multicast stream is started only when the first customer joins it.

The probability density function of inter-arrival time x is given by

$$f_{A_n}(x) = \sum_{i=1}^{n} P_i \lambda_i e^{-\lambda_i x}, \ x \geq 0 \quad \dots(3)$$

where A_n is a hyper-exponential random variable with n exponential stages and is the probability that A_n will take on the form of the exponential distribution with rate λ_i such that $\sum_{i=1}^{n} P_i = 1$.

The average number of streams can be computed as

$$N = \frac{\int_0^{w-1} B * f_{A_n}(x)dx}{C*W} \quad \dots(4)$$

$$= \frac{\int_0^{w-1} [Cx\beta(\beta+1) + (L\text{-}2x)C] \, f_{A_n}(x)dx]}{C*W} \quad \dots(5)$$

$$= \frac{\beta(\beta+1) \text{-}2) \int_0^{w-1} x f_{A_n}(x)d(x) + L \int_0^{w-1} f_{A_n}(x)dx}{W} \quad \dots(6)$$

In the next section we find out the number of streams corresponding to different batching times and the optimal batching time at different arrival rates of customers.

4.0 ADAPTIVE BATCHING SCHEME

Adaptive batching scheme is used to calculate the average number of streams required for multicasting. In this scheme, the batching time does not remain fixed and depends on the average number of requests. Here, we have used a pseudocode given below to select the optimal number of streams. The optimal

Table 1: Optimal batching time (W) for various arrival rates

$p_1=p_2=0.05$				$p_1=1,p_2=0$			
L=150		L=120		L=150		L=120	
Arrival /minute (l) (λ) $\leq$	Optimal batching time (W minutes)	Arrival /minute (λ) $\leq$	Optimal batching time (W minutes)	Arrival /minute (λ) $\leq$	Optimal batching time (W minutes)	Arrival /minute (λ) $\leq$	Optimal batching time (W minutes)
0.02	18	0.02	17	0.02	18	0.02	18
0.03	14	0.03	16	0.022	16	0.023	16
0.04	12	0.04	15	0.023	15	0.025	15
0.05	11	0.05	14	0.026	13	0.027	14
0.06	10	0.06	13	0.028	10	0.03	13
0.07	9	0.07	12	0.38	9	0.035	11
0.1	8	0.08	11	0.45	7	0.04	10
0.2	6	0.09	10			0.041	9
0.3	5	0.1	9			0.43	8
0.4	4	0.2	8			0.045	6
0.5	3	0.3	7			0.05	5
0.7	2	0.5	6				
		0.8	5				

number of streams is calculated for different arrival rates and the length of the movie. Table 1 gives the optimal batching time for various arrival rate for different durations of movies. When the first request arrives, the time for opening the most recent multicast stream t_x is set as t_x=-2W. The arrival rate of customers (L) is updated. Initial number of streams is calculated by equation 4, which is updated according to the time required for synchronization of multicast and the unicast stream (t_s), and the buffer size (b). A new multicast stream is opened if time of synchronization is greater than the batching time (t_s>W) and again, the t_x is updated. A new multicast stream is also opened when the synchronization time is greater than the buffer size because at that time, the buffer is fully exhausted. The pseudocode for the adaptive batching stream is given below.

Input: λ, L, W // arrival rate, length of movie and batching time respectively
Output: N // number of multicast streams.
For every new arrival

Update L // average arrival rate
Compute t_c // arrival time of customer
Locate corresponding W// batching time
Compute N // by using equation 4
Compute $t_s = t_c - t_x$
    If $t_s > W$
    N=N+1 // open a new multicast stream
    $t_x = t_c$ // Update t_x
else
    if $t_s > b$
        N=N+1
        $t_x = t_c$
    else
        Open a dedicated stream with double transmission rate.
Return N

5.0 NUMERICAL ILLUSTRATIONS

Both the fixed batching scheme and the adaptive batching schemes are compared using numerical illustrations with the help of MATLAB 6 software. As a special case, we computed the results for n=2; but the results held equally good for higher values of n. For illustration purposes, we took b=10, $l_2=2l_1$, so that $l=3l_1$. Figures 1(a) and 1(b) show the variation of N with respect to W for various values of l_1 while taking L=120. It can be noticed that the number of streams are less for lower arrival rates and are more for higher arrival rates. The optimal

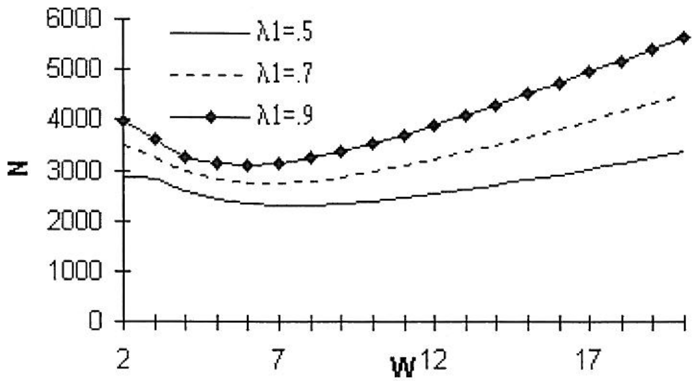

Fig. 1 (a): Number of streams vs batching time for $p_1=p_2=0.05$

batching time for various arrival rates are exhibited in Table 1. A greater number of streams is required for movies of longer duration for both fixed and adaptive batching schemes as depicted by Figures 2(a) and 2(b) respectively. Similarly, bigger number of streams is required for higher batching time in fixed batching scheme as shown in Figure 2 (c). Figure 3 (a) reveals that for higher arrival rates, the number of streams required by adaptive batching scheme, is less than the fixed batching scheme. Table 2 displays the percentage reduction in the number of streams by adaptive batching strategy for various arrival rates using three different durations of movies. It can be seen that for movies of shorter duration (L=90), up to 57% reduction can be achieved in the number of streams whereas for L=150, the number of streams can be reduced up to 41%. Hence, we can see that adaptive batching scheme considerably reduces the number of streams.

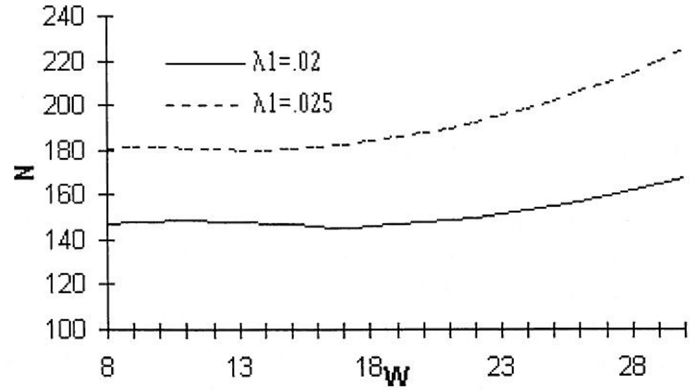

Fig. 1 (b): Number of streams vs batching time for p_1=1, p_2=0

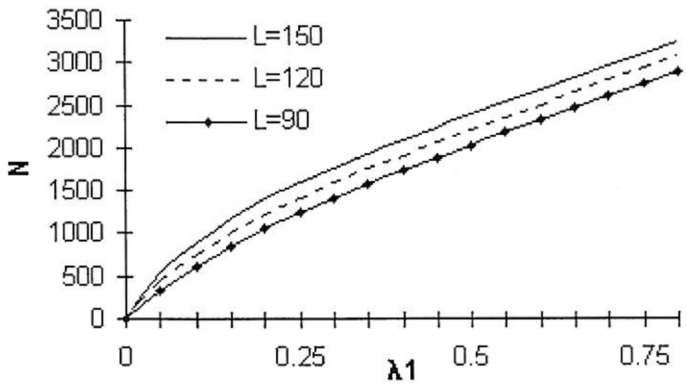

Fig. 2 (a): Number of streams vs arrival rates for different values of L (FBS)

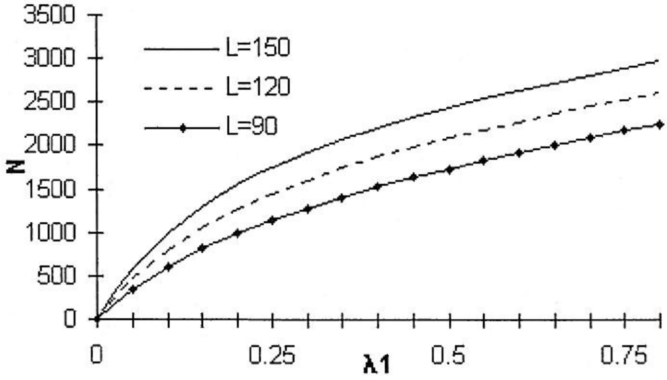

Fig. 2 (b): Number of streams vs arrival rates for different values of L (ABS) taking $p_1=p_2=0.05$, L=120

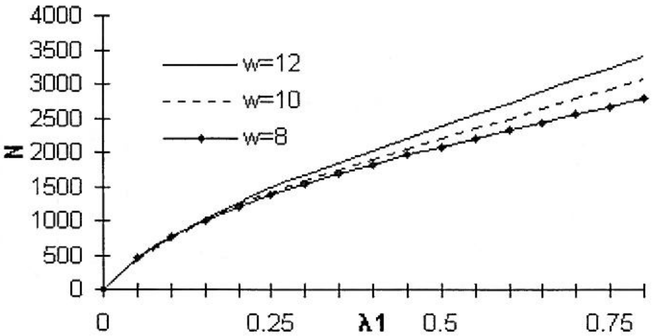

Fig. 2 (c): Number of streams vs arrival rates for different values of W (FBS) taking $p_1=p_2=0.05$, L=120

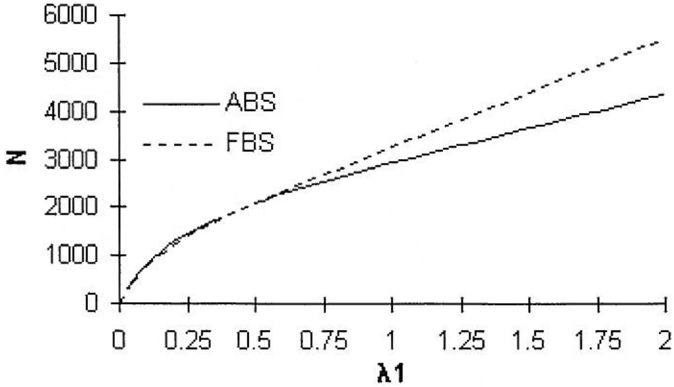

Fig. 3: Comparison of FBS and ABS with respect to N for L=120 minutes taking $p_1=p_2=0.05$

Table 2: Percentage reduction in the number of streams using adaptive batching scheme (p_1=p_2=0.05)

	L=150			L=120			L=90		
	N	N		N			N	N	
l	FBS	ABS	% Reduction	FBS	ABS	% Reduction	FBS	ABS	% Reduction
0.1	992.04	901.29	9.15	797.77	753.22	5.59	603.51	605.14	0.27
0.2	1,555.45	1399.79	10.01	1272.59	1225.95	3.66	989.72	1052.11	6.30
0.3	1,932.04	1771.07	8.33	1608.49	1592.44	1.00	1284.94	1413.82	10.03
0.4	2,213.05	2089.65	5.58	1870.62	1910.02	2.11	1528.20	1730.40	13.23
0.5	2,440.00	2388.40	2.11	2088.70	2208.52	5.74	1737.40	2028.64	16.76
0.6	2,634.72	2679.48	1.70	2279.16	2499.52	9.67	1923.61	2319.57	20.58
0.7	2,809.83	2967.19	5.60	2452.19	2787.20	13.66	2094.55	2607.22	24.48
0.8	2,972.98	3253.20	9.43	2614.28	3073.21	17.55	2255.58	2893.21	28.27
0.9	3,128.77	3538.29	13.09	2769.51	3358.29	21.26	2410.26	3178.29	31.87
1	3,280.01	3822.85	16.55	2920.45	3642.85	24.74	2560.89	3462.85	35.22
1.1	3,428.37	4107.10	19.80	3068.64	3927.10	27.98	2708.91	3747.10	38.33
1.2	3,574.87	4391.17	22.83	3215.04	4211.17	30.98	2855.21	4031.17	41.19
1.3	3,720.13	4675.12	25.67	3360.24	4495.12	33.77	3000.35	4315.12	43.82
1.4	3,864.52	4958.98	28.32	3504.59	4778.98	36.36	3144.66	4598.98	46.25
1.5	4,008.30	5242.79	30.80	3648.35	5062.79	38.77	3288.39	4882.79	48.49
1.6	4,151.63	5526.56	33.12	3791.66	5346.56	41.01	3431.69	5166.56	50.55
1.7	4,294.63	5810.29	35.29	3934.65	5630.29	43.10	3574.67	5450.29	52.47
1.8	4,437.37	6094.00	37.33	4077.39	5914.00	45.04	3717.40	5734.00	54.25
1.9	4,579.92	6377.68	39.25	4219.93	6197.68	46.87	3859.94	6017.68	55.90
2	4,722.32	6661.35	41.06	4362.33	6481.35	48.58	4002.34	6301.35	57.44

6.0 CONCLUSION

VOD is becoming an important area of interest due to increasing demand of customers for multimedia applications. In this paper, we have suggested an adaptive batching scheme by assuming hyper-exponential distribution for arrival of customers. We have compared the adaptive scheme with the fixed scheme of batching. It has been shown that adaptive batching strategy uses a smaller number of multicast streams as compared to the fixed batching policy; hence it can be used for optimum utilization of bandwidth thereby reducing the overall cost. Though we provided the numerical results for a special case of the hyperexponential distribution, the results are equally good for the general case. Companies offering batching services for multimedia applications can utilize the suggested scheme. Our work can be further extended on the analysis of efficient utilization of resources for VOD servers.

REFERENCES

Aggarwal, C., Wolf, J., & Yu, P.S. (1996). On optimal batching policies for video-on-demand storage servers. *IEEE Proceedings of Multimedia'96,* 253-258.

Chen, S. & Thapar, M. (1997). A novel video layout strategy for near video-on-demand servers. *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, 37-45.

Dan, A., Sitaram, D., & Shahabuddin, P. (1994). Scheduling policies for an on demand video server with batching. *Proceedings of the ACM Multimedia, San Francisco*, 391-398.

Feldmann, A. & Whitt, W. (1998). Fitting mixtures of exponentials to long tailed distributions to analyze network performance models. *Performance Evaluation, 31*, 245-479.

Gao L. & Towsley D. (1999). Supplying instantaneous video-on-demand services using controlled multicast. *IEEE International Conference on Multimedia Computing and Systems*, *2*, 117-121.

Gemmel D. J., Vin H. M., Kandlur D. D., & Rangam P. V. (1995). Multimedia storage servers: A tutorial and survey. *IEEE Computers, 28*(5), 40-49.

Hua K. A., Cai Y., & Sheu S. (1998). Patching: A multicast technique for true video-on-demand services. *Proceedings of 6th ACM Intelligent Multimedia Conference,* 191-200.

Jain, M. & Kriti, P.(2003). Batching strategy for multicasting in video on demand (VoD) system. *International Computer Science, 6*(1), 1-9.

Kim, H. J. & Zhu, Y.(1998). Channel allocation problem in VOD system using batching adaptive piggybacking. *IEEE Transaction on Consumer Electronics, 44*(3), 969-976.

Poon, W.F. & Lo, K. T. (1999). New batching policy for providing true video-on-demand (T-VoD) in multicast system. *Proceedings of IEEE International Conference on Communications (ICC'99), 2*, 983-987.

Poon, W.F., Lo, K.T., & Feng, J. (2001). Adaptive batching scheme for multicast video-on-demand systems. *IEEE Transaction on Broadcasting, 47*(1), 66-69.

Porhuri, S., Petr, D.W., & Khan, S. (2002). Characterizing and modeling network traffic variability. *IEEE Conference on Communications*, 2405-2409.

Sheu S., Hua K. A., & Hu T. (1997). Virtual batching: a new scheduling technique for video-on-demand servers. *Proceedings of the 5th International Conference on Database systems for Advanced Applications,* 481-490.

Wong, W.T., Zhang, L., & Pang, K.K. (1995). Video on demand service policies. *Proceedings of the IEEE Singapore International Conference on Networks*, 560-564.