Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp815-821)

An Adaptive Hybrid Video-on-Demand System

Chenn-Jung Huang, Yi-Ta Chuang
Institute of Learning Technology
National Hualien University of Education
Hualien, Taiwan 970

Wei Kuang Lai, Hsin Hung Sung
Department of Computer Science and Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan 804

Abstract: - As streaming video and audio over the Internet become popular, the deployment of a large-scale multimedia streaming application requires an enormous amount of server and network resources. In a Video-on-Demand (VoD) environment, batching of video requests are often used to reduce I/O demand and improve throughput. Since users may leave if they experience long waits, a good video scheduling policy needs to consider not only the batch size but also the user defection probabilities and waiting times. Besides, a practical VoD resource sharing scheme should try its best to provide some free stream to serve a high priority client's request immediately because the high priority clients might pay for the requested video. To tackle the above problems, this work proposes a hybrid resource sharing model which combines controlled multicasting and batching scheme. A bandwidth borrowing and reserving scheme is adopted in our hybrid model to give high priority clients prompt service whereas provide low priority clients comparable service as given by the representative scheduling policies in the literature. The experimental results demonstrate that our proposed resource sharing scheme is effective and feasible when blocking probability of high priority clients and defection probability of low priority users are used as the performance metrics.

Key-Words: - video-on-demand, batching, bandwidth borrowing and reserving, scheduling, controlled multicasting, quality of service.

1 Introduction

Recent advances in communication and computer technology have made transmission rate of the Internet faster and faster. Next-generation networks will support transmission rate that are orders of magnitude higher than current rates. Because of the deployment in the Internet, the explosive increase in commercial usage of the Internet has resulted in a rapid growth in demand for video deliver technologies. In such a system that is implemented with client-server architecture, viewers have the flexibility of choosing both the video they want as well as the time at which they wish to watch the video.

Batching is the most general policy for VoD system which groups users waiting for the same video data and then serves them using a multicast channel. This batching process can occur passively while the users are waiting or actively by delaying the service of early-arriving users to wait for late-arriving users to join the batch.

Controlled multicasting [1] is a reactive instantaneous VoD system. Assuming the bandwidth that the server provides is unlimited and the clients' buffer size is not a constraint, the number of video streams required in this scheme is $O(\sqrt{L\lambda})$ on average, where L is the video length and λ is the user request arrival rate. The advantage of controlled multicasting is that earlier request can be served instantaneously without waiting later request as required in the batching scheme.

This paper presents a hybrid system which combines controlled multicasting and batching schemes to benefit from the advantages of both schemes. A bandwidth borrowing technique is also employed in this work to handle the temporary bandwidth crisis in original controlled multicasting scheme. Besides, a bandwidth balanced scheduling policy is proposed to improve the performance of our VoD system.

The remainder of this paper is organized as follows. Section 2 provides a brief overview of the resource sharing schemes and the scheduling policies used in the batching scheme. In Section 3, the hybrid resource sharing scheme wherein a channel borrowing and reserving mechanism are embedded is presented. The simulation result is given in Section 4. Conclusion is made in section 5.

2 Related works

2.1 Resource sharing and scheduling

It is well known that the number of clients supported by a VoD server is highly constrained by the requirements of real-time playbacks and the high transfer rates. Thus, a wide spectrum of techniques

was developed to enhance the performance of VoD servers, including resource sharing and scheduling [2,3], admission control, disk striping, data replication, disk head scheduling, and data block allocation and rearrangement. This work restricts the discussion on resource sharing and scheduling where the server uses multicast stream to serve the clients who request the same video.

The performance of VoD servers can be significantly improved through resource sharing. The categories of resource sharing strategies include batching [2], patching, piggy-backing and broadcasting [3]. In batching, requests to the same movies are accumulated and served simultaneously. Patching expands the multicast tree dynamically to include new requests and reduces the request waiting time with the expense of additional bandwidth and buffer spaces needed at the client's site.

Piggy-backing services a request almost immediately but adjusts the playback rate so that the request can catch up a preceding stream, resulting in a lower-quality initial presentation. For broadcasting scheme, a video is fragmented into a number of segments. Each segment is periodically broadcasted on a dedicated channel. Broadcasting requires relatively very high bandwidth and buffer spaces at the clients' sites.

2.2 Scheduling policies

The performance of the above-mentioned resource sharing strategies can be further improved if VoD servers can schedule the waiting requests in an appropriate order. The difference among the various scheduling policies in resource sharing approach is the policy to select which batch to serve first when a server channel becomes available. Fig. 1 depicts three static multicast schemes. In first-come-first-serve (FCFS), as soon as the bandwidth for some server becomes available, the batch holding the oldest request with the longest waiting time is served immediately. In maximum-queue-length-first (MQL) [2], the batch with the largest number of pending requests (i.e., longest queue) is chosen to receive the service. FCFS offers fairness since the scheme treats each user equally regardless of the popularity of the requested video. This scheme, however, yields low system throughput because it may choose to serve a batch with fewer requests first while cause another batch with more requests to wait. To address this issue, MQL, which also maintains a separate waiting queue for each video, delivers the video with the longest queue (i.e., the largest number of pending requests) first. This policy maximizes server throughput, but is unfair to the users who request less popular videos. Maximum-factored-queued-length first (MFQL) [4] attempts to provide reasonable fairness as well as high server throughput. This scheme also maintains a waiting queue for each video. When a server channel becomes free, MFQL selects the video v_i with the longest queue weighted by a factor $1/\sqrt{f_i}$ to deliver, where f_i denotes the access frequency or the popularity of the video v_i. The factor f_i prevents the server from favouring the popular videos at all times. However, it was observed that MFQL is not fair in most situations because it is sorely determined by the queue length of the video. It is well known that popular movies always have the longer queue length than the others, and the effect of the factor f_i is much smaller than the queue length.

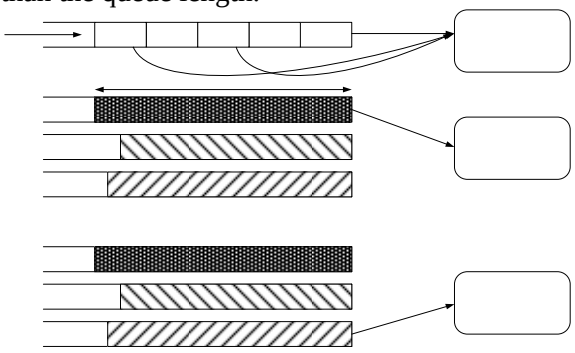

Fig.1. FCFS, MQL, and MFQL models.

3 The Hybrid Video-on-Demand System

3.1 Overview of proposed scheme

In the primitive VoD systems, one of the above mentioned policies is simply employed into the VoD schemes. For periodic broadcasting scheme, this policy can be only justified for very popular videos. In batching scheme, early-arriving users are punished by waiting for late-arriving users. For patching scheme, it requires additional client-side's bandwidth and buffer space. As for piggy-backing, it exploits users' tolerance on playback rate variation. This work thus proposes an adaptive hybrid system, which includes controlled multicasting and batching schemes, to tackle the drawbacks exhibited in the four above-mentioned schemes.

3.2 Integration of controlled multicasting and channel borrowing

Controlled multicasting [1] scheme assumes that there are infinite counts of server channels. This scheme allows two clients that request the same

video whereas arrived at different time period to share a channel. However, the late-arriving client is allowed to use another free channel to download the portion of the video segment that was received by the early-arriving client. The difference between this scheme and other batching schemes is that the controlled multicasting does not delay the earlier request while still possesses the advantage of channel sharing.

Channel borrowing scheme [5] was proposed to handle the temporary bandwidth crisis on controlled multicasting by borrowing bandwidth from other ongoing streams. Similar to that defined in coding standards as MPEG-2, video streams in this scheme are coded into multiple layers which comprise base layer and several enhanced layers using scalable layer video coding technique, and each layer corresponds to a specified QoS. During a temporary bandwidth crisis, the topmost of some of the ongoing video streams are removed to accommodate new stream admission. When bandwidth becomes available after regular or patch stream is dropped, the missing layers of the streams are restored to reuse that free bandwidth.

This work first divides clients into two different priority groups based on their certification guarantees. The clients in different classes have different QoS guarantees and only the high priority clients need to pay for the requested videos. Thus the controlled multicasting scheme is solely dedicated for the client in high priority group, whereas batching scheme is employed to serve low priority clients that access the videos for free. Each video is divided into several parts, and the approach taken by the broadcasting scheme is adopted in this work. That is, each channel is divided into fixed time slots, and each part of the videos is transmitted in the designated time slot. The videos in the two priority groups are broadcasted in turn. The controlled multicasting and batching scheme are integrated in this work by means of changing the mode in each channel after fix time slot. For high class clients, they have higher priority to access channels in controlled multicasting mode, whereas the low class clients are preferred to access channels in batching mode first. The low class clients can access channels in controlled multicasting mode in case the current network loading is not heavy. Channel borrowing technique is used to allow high priority clients to share a channel that is currently serving low class clients by means of removing the topmost layer of the ongoing streams which serves low class clients in controlled multicasting mode. The shared channel is used to catch up the missing video segment received by the early-arriving high priority clients. The difference between our scheme and broadcasting scheme is in broadcasting scheme the videos are broadcasting in turn which is only suitable for popular videos and consumes more server resources while in our scheme the access mode of the channel is changed in turn which has more flexibility than broadcasting and can take the advantage both on controlled multicasting and batching mode.

3.3 Adaptive channel reserving mechanism

An adaptive channel reserving mechanism is implemented in this work to reserve some free channels for the incoming high class clients to ensure their higher priority and lower their blocking probability.

Since our hybrid resource sharing scheme is anticipated to reserve an appropriate number of free channels for the expected incoming high priority clients during the next time period, this work thus employs Eq. (1) to determine the number of the reserved channels for the high class clients during the next time period based on the current network traffic load,

$$Rr(t+1) = K \cdot Str \cdot \frac{b(t) \cdot \hat{n}_H(t+1)}{\hat{n}_L(t+1)} \qquad (1)$$

where Str denotes the total number of the server streams, b is the blocking probability for class 1 video during the current time period, $\hat{n}_H(\cdot)$ and $\hat{n}_L(\cdot)$ represent the predicted number of high priority clients and that of low priority clients during next time period, respectively, and K is a constant smaller than 1. Notably, the weighted moving average method is used to predict the values of $\hat{n}_H(\cdot)$ and $\hat{n}_L(\cdot)$.

3.4 Scheduling policy with bandwidth balancing (SPBB)

Two approaches that handle multi-priority traffic were proposed in distributed-queue dual-bus (DQDB) networks with and without bandwidth balancing [6].

In the so-called “local” approach, bandwidth balancing procedure guarantees that there is some unused bus capacity and each parcel is asked to restrict its throughput to some multiple of that spare capacity. However, the proportionality factor depends on the priority level of the parcel. Specifically, the parcel of priority p is asked to restrict its throughput to a multiple M_p of the spare bus capacity; parcels with less demand than this may acquire all the bandwidth they desire. Note that every active parcel in the network gets some bandwidth.

Parcels of different priority levels are provided with the bandwidth in proportion to their bandwidth balancing moduli M_p. Given the offered loads $\rho_p(n)$ and the bandwidth balancing moduli M_p, the carried loads $r_p(n)$ can be obtained accordingly. In the special case where all parcels of priority level p, N_p, have heavy demand, the solution turns out to be an especially simple form:

$$r_p(n) = \frac{M_p}{1 + \sum_q M_q \cdot N_q} \tag{2}$$

In the so-called "global" approach, it assumes that every node can determine the bus utilization due to traffic of each priority level. Each parcel is asked to limit its throughput to some multiple M of the spare bus capacity not used by parcels of equal or greater priority; parcels with less demand than this may have all the bandwidth they desire. Given the offered loads $\rho_p(n)$ and the bandwidth balancing modulus M, the carried loads $r_p(n)$ is derived subsequently. In the special case where all N_p parcels of priority level p have heavy demand, the solution has a simple form:

$$r_p(n) = \frac{M}{\prod_{q \geq p} (1 + M \cdot N_q)} \tag{3}$$

It can be seen that the above two approaches always preserve some bandwidth for low priority clients. In this work, the low priority clients as mentioned in the preceding subsection are further divided into two different priority subgroups based the popularity of the videos, and the bandwidth balancing mechanism is employed in our proposed scheduling policy to organize the playing order of the videos in batching scheme. The main goal is to benefit fairness from bandwidth balancing mechanism by allowing the clients that request the less popular videos to receive the service, and hopefully achieve better performance with the scheduling policy in case a suitable scheduling policy is found. Meanwhile, it is expected that the hot videos have more service times than cold videos. In our scheduling policy, two different popularity classes are defined as P_1 =3 and P_2=1. The VoD server will select the videos in class P_1 to serve first, and continuously serve different videos in class P_1 for next two time periods. Then the server will select videos in class P_2 to serve next. When completing serving the class P_2 client, the server will again select videos in class P_1 to serve for the next three time periods.

Since there are different videos waiting for service in the same priority class in the batching scheme, another selecting mechanism is needed to select which video in the same priority group to serve next. The formula used in Largest Aggregated Waiting Time First (LAW) policy [7] is modified in this work. The choice of the video is determined by the queue length and the clients' aggregated waiting time. LAW considers the arrival time of each request which is ignored by MFQL. This factor makes LAW fairer than MFQL. When a stream becomes available, LAW policy schedules the video with the largest S_i as given in the following equation,

$$S_i = (t \cdot n - \sum_{m=0}^{n} a_{im}), \tag{4}$$

where a_{ij} denotes the arrival time of request j on video i, n is the total number of request for video i, and t represents the current time.

The unfairness issue is further considered in this work and the LAW formula is modified as follows:

$$V_i = (t \cdot n - \sum_{m=0}^{n} a_{im}) \cdot \sqrt{\alpha_i}\ , \tag{5}$$

where α_i represents the accumulated time of video i since last service.

The VoD server selects the video to serve according to its V_i value. The video having the biggest V_i value will be served first. By adding the factor α_i into our scheduling policy, it is hoped that the defection probability that the low priority users leave the VoD system can be lowered and the unfairness can be improved.

4. Simulation

A series of simulations were conducted to compare our proposed schemes (AHVoD) with FCFS and MFQL. The parameters used in the simulations are summarized in Table 1.

4.1 Performance metrics

In the analysis of our resource sharing and scheduling policies, the following performance measures are used:

- Blocking probability: This is the probability that an arriving high priority client leaves the system without being serviced due to the lack of server stream.
- Defection probability: This is the probability that an arriving low priority client leaves the system without being served due to the waiting time exceeding the viewer's tolerance. Obviously, the defection probabilities may vary across different videos. Let r_i denote the defection probability for video i, then the mean defection probability can be expressed by,

$$\bar{r} = \sum_{i=1}^{N} r_i / N \cdot \quad (6)$$

- Average latency time: The latency of a client is the period which elapses between the arrival of the video request and the time when the service to the display device is actually initiated. Only non-defecting clients are considered in the latency time measure.

Arrival rate	20-50 requests/per minute
Frequency	Zipf-like distribution
Video number	100
Batch time	5 minutes
Video length	120 minutes
Simulation time	8 hours
Priority level	2
Server capacity	100-500 streams

Table 1. Simulation parameters

4.2 Simulation result

A series of experiments were conducted wherein arrival rate was varied from 20-60 requests per minute and server capacity varied from 100-500 streams.

Figures 2 to 4 show the simulation results where the arrival rate was varied from 20 to 60 requests per minute. The server capacity is fixed at 200 streams. It can be seen from Fig 2 that the proposed hybrid system embedded with channel borrowing and reservation mechanisms (AHVoD3) achieves better performance than the hybrid scheme embedded with channel borrowing mechanism (AHVoD2) and the primitive hybrid scheme (AHVoD1). Meanwhile, the hybrid embedded with channel borrowing mechanism alone also significantly outperforms the primitive hybrid scheme. Thus the channel borrowing and adaptive channel reservation mechanism employed in this work indeed boost the performance of the proposed resource sharing system.

The defection probability of low priority clients for the three proposed schemes and the representative scheduling scheme, MFQL, is given in Fig. 3. Notably, the primitive hybrid scheme (AHVoD1) performs worse than MFQL scheme because the prompt service of high priority clients in controlled multicast operating mode deteriorates the defection probability of low priority clients operated in batching mode. However, the channel borrowing and reserving mechanisms effectively rectify the deterioration problem as given in Fig. 3.

The service delay time of low priority clients for the three proposed schemes and MFQL is given in Fig. 4. The three proposed schemes slightly achieve better performance than MFQL.

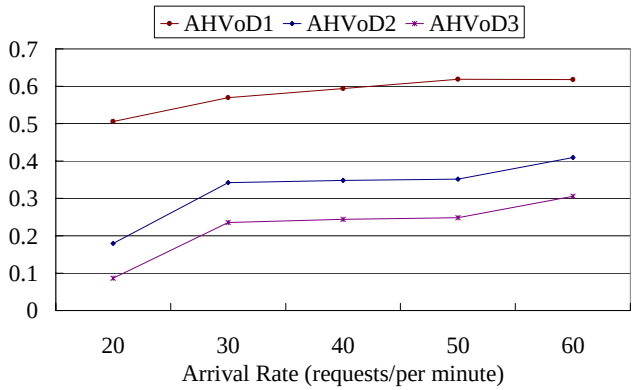

Fig.2 Blocking probability of high priority clients for the three proposed schemes at varied arrival rates.

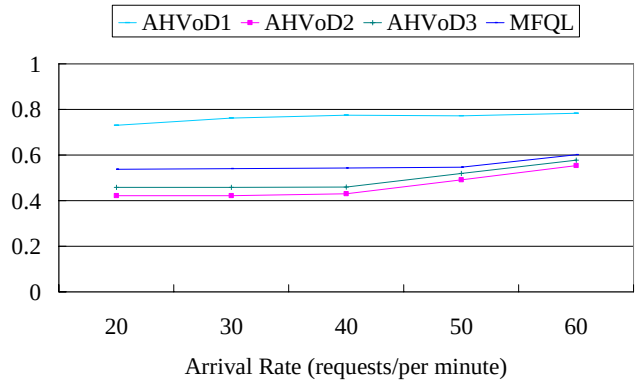

Fig.3 Defection probability of low priority clients at varied arrival rates.

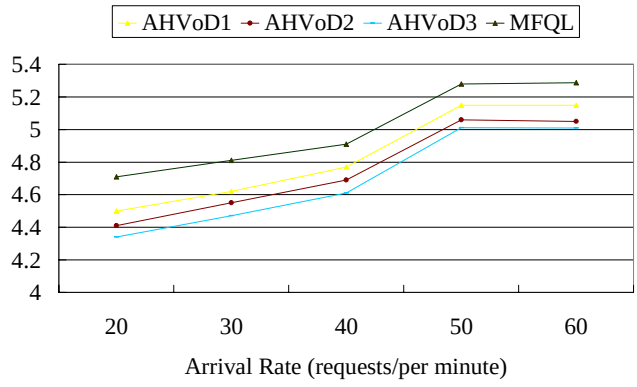

Fig.4 Service delay time of low priority clients at varied arrival rates.

Figures 5 to 7 show the simulation results with the server capacity varied from 100 to 500 streams. The arrival rate is fixed at 50 requests per minute. As illustrated in Figs. 5 and 6, the channel reserving mechanism can further lower the blocking probability of high priority clients whereas slightly degrade the defection probability of low priority clients when the server capacity is less than or equal

to 200. The service delay time for the three proposed schemes is smaller than that of MFQL as given in Fig. 7.

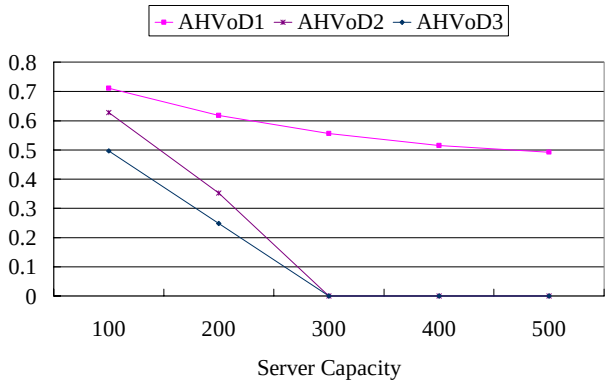

Fig.5 Blocking probability of high priority clients for the three proposed schemes with the varied server capacity.

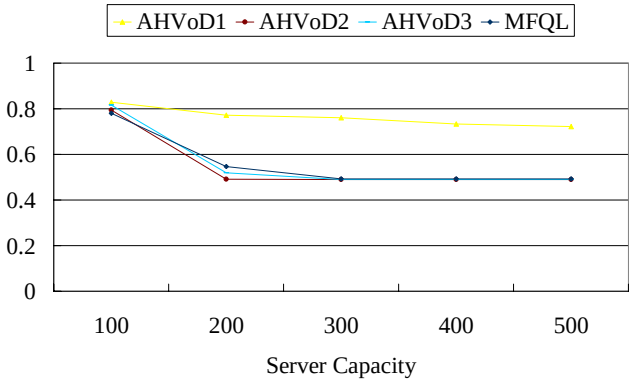

Fig.6 Defection probability of low priority clients with the varied server capacity.

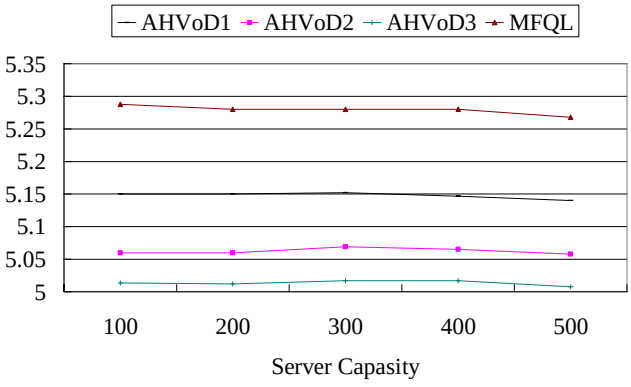

Fig.7 Service delay time probability of low priority clients with the varied server capacity.

5. Conclusion

In this work, the high priority clients that pay for the requested videos are allowed to quickly receive the service by the VoD server, whereas permit the users to access the video for free to have opportunity to be served as well. A hybrid VoD resource sharing system along with bandwidth borrowing technique was proposed to effectively lower defection probability and the waiting time for the clients in different priority groups. Furthermore, an adaptive bandwidth reserving mechanism is also presented in this work to let the VoD server be more efficient in the usage of the free streams. A series of simulations were conducted to compare the proposed hybrid VoD system with the well-known MFQL scheduling schemes. It was observed that the proposed hybrid scheme effectively lower the blocking probability of high priority clients and performs better than MFQL in terms of defection probability and delay latency of the low priority clients. Thus the superiority and the feasibility of the proposed hybrid scheme is verified.

6. Acknowledgment

The authors would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC 94-2213-E-026-001.

References

[1] L. Gao and D. Twosly, “Supplying instantaneous video-on-demand services using controlled multicast” *Proceedings of IEEE International Conf. on Multimedia Computing and Systems*, 1999, 117-121.

[2] A. Dan, D. Sitaram, and P. Shahabuddin, “Scheduling Policies for an On-Demand Video Server with Batching” *Proceedings of the ACM Conf. on Multimedia*, 1994, 391-398.

[3] L. Juhn and L. Tseng, “Harmonic Broadcasting for Video-on-Demand Service,” *IEEE Trans. on Broadcasting*, 43(3), 1997, 268-271.

[4] C. C. Aggarwal, J. L.Wolf, and P. S. Yu, “The Maximum Factor Queue Length Batching Scheme for Video-on-Demand Systems” *IEEE Trans. on Computers*, 50(2), 2001, 97-110.

[5] S.A. Azad, M. Murshed and L.S Dooley, “Bandwidth borrowing schemes for instantaneous video-on-demand systems,” *IEEE International Conference on Multimedia and Expo*, 3, 2004, 2011-2014

[6] E.L. Hahne, A.K. Choudhury, N.F Maxemchuk, “DQDB networks with and without bandwidth balancing” *IEEE Transactions on Communications* 40(7), 1992, 1192-1204.

[7] K.A. Hua,, Oh Junghwan and Vu Khanh, “An adaptive hybrid technique for video multicast,” *Proceedings of 7th International Conference on*

Computer Communications and Networks, 1998,
227-234