Bandwidth Borrowing Schemes for Instantaneous Video-on-Demand Systems

Salahuddin A. Azad, Manzur Murshed and Laurence S. Dooley
Gippsland School of Computing and IT, Monash University, Churchill Vic 3842, Australia
{Salahuddin.Azad, Manzur.Murshed, Laurence.Dooley}@infotech.monash.edu.au

Abstract

Controlled multicast scheme provides instantaneous service, but limited server bandwidth causes some user requests to be either delayed or rejected when insufficient free bandwidth is available. In this paper, two borrowing schemes are proposed for instantaneous video-on-demand (VOD) that reduce user request blocking rate by borrowing bandwidth from ongoing video streams when there is insufficient free bandwidth for the server to deliver a new video stream. Both these new schemes have been proven to successfully reduce blocking rate and increase bandwidth utilization at the expense of temporarily degrading the video quality.

1. Introduction

Video-on-demand (VOD) provides a mechanism for watching a selected video at any time, independent of the choice of other viewers. In *true-VOD* (TVOD) systems, a viewer is served instantaneously by being allocated a dedicated unicast channel. At high client request rates, the TVOD server bandwidth requirement however, can become a severe bottleneck. *Near-VOD* (NVOD) or *quasi-VOD* (QVOD) systems reduce the server bandwidth requirement significantly by forcing one or more clients to share the same video stream. This provides scalability at the expense of introducing user delay so the client may have to wait for a certain amount of time before the requisite video stream starts.

There are two main approaches to providing NVOD services—*reactive* [1]-[7] and *proactive* [8]-[12] protocols. In the former, the server allocates a channel for a video only when there are requests from clients for that video via an upstream channel. In this *client initiated* system, the server bandwidth requirement becomes substantially high when the arrival rate is high. In proactive protocols, the video is periodically broadcast into several fixed channels. By keeping the number of channels independent of the number of clients, this scheme can theoretically serve an unlimited number of clients simultaneously within a bounded client delay. This *server initiated* broadcasting also eliminates the need to handle individual client request through an upstream channel. However, periodic broadcasting can only be justified for very popular videos.

Controlled multicasting [5] is a reactive instantaneous VOD system. The number of video streams required in this scheme when server bandwidth is unlimited and client buffer size is not a constraint, is on the average $O(\sqrt{L\lambda})$, where L is the video length and λ is the user request arrival rate. In a finite bandwidth server however, some users will be blocked due to arrival rate variance. Blocking can be avoided by allocating excess bandwidth which causes lower utilization of server bandwidth. An alternative to blocking a user's request is to delay the request until a video stream becomes free; however this is undesirable in instantaneous VOD services. Such schemes also do not provide any upper bound on the user delay using a fixed amount of server bandwidth. Moreover, fixed bandwidth proactive schemes can provide more efficient service at the expense of user delay when the arrival rate is moderate and high.

In this paper, the temporary bandwidth crisis in controlled multicasting is overcome by borrowing bandwidth from other ongoing steams. To facilitate such borrowing, video streams are coded into multiple layers (similar to that defined in coding standards such as MPEG-2) comprising one base layer and several enhancement layers using *scalable layered video coding*, where each layer corresponds to a particular QoS. During a temporary bandwidth crisis, the top-most layers of some of the video streams are removed to accommodate new stream admission. After removing the topmost layers of all the ongoing streams, the next topmost layers are removed. When bandwidth becomes available after a regular or patch stream is dropped, the missing layers of the streams are restored using that free bandwidth. Thus user request blocking is reduced by temporarily degrading the quality of video steams. The rest of the paper is organized as follows. Section 2 briefly reviews some related works. Section 3 presents the proposed bandwidth borrowing schemes, while section 4 presents the theoretical basis behind the schemes. Section 5 discusses the simulation results and section 6 concludes the paper.

0-7803-8603-5/04/$20.00 ©2004 IEEE

2. Related works

The *Borrow-and-return* scheme proposed in [13] reduces user waiting time significantly by borrowing bandwidth in proactive VOD systems. This scheme utilizes the unused time slots of videos without viewers to those videos with viewers to speed up latter's transmission and thus reduces the viewer's waiting time. The effectiveness of their scheme was demonstrated by applying it to *fast broadcasting* scheme. This scheme performs very well for low and moderate arrival rates but at higher arrival rates its performance deteriorates significantly.

3. Proposed borrowing schemes

The bandwidth borrowing schemes presented in this paper are different to that in [13]. They are reactive instantaneous VOD systems that aim to reduce blocking rate rather than waiting time. Since scalable layered video coding is considered in these schemes, the borrow-able amount is always fixed. In other words, if the top layer of a video stream has *bw* bandwidth, then either *bw* bandwidth is borrowed at a time or nothing is borrowed. This means bandwidth may not be equally borrowed from all channels. Appropriate channels therefore need to be identified from which to borrow when there are more available channels than necessary. This choice should be that as few a number of users are affected and the quality of different ongoing streams is kept as close as possible. If the bandwidth of a stream is reduced then the stream quality is said to be *degraded* and conversely, if the bandwidth of a stream is increased then the stream quality is said to be *upgraded*.

In controlled multicast, patch streams have relatively short duration and dedicated to one user only, so borrowing from these streams would have much less impact on users than borrowing from the regular ones that have relatively long duration and used by many users. Therefore in case of bandwidth crisis the topmost layers of the patch streams would be borrowed first. Once the topmost layers of all the patch streams are removed, then the topmost layers of the regular streams are borrowed. After removing the topmost layers of all the streams, the next topmost layer would be selected for borrowing. Obviously, streams having the minimum QoS cannot be chosen for borrowing. Bandwidth is allocated for new stream admission by borrowing as much as possible but no new stream can have more bandwidth than any of the ongoing streams. If it is found that there is not enough borrow-able bandwidth to admit new stream with the minimum QoS then no borrowing is done and the user request is blocked.

When free bandwidth becomes available after a regular or patch stream is dropped, the lowest missing layers of the regular streams are restored first. Once these missing layers are restored for all the regular streams, then the lowest missing layers of the patch streams are restored. Next all enhancement layers are candidates for restoration. In fact, the scheme has a hierarchical strategy that facilitates both graceful degradation and enhancement of video quality. It ensures higher video quality is observed by as many users of possible whilst the borrowing affects as few users. It also is ensured that the contrast of quality between different streams is minimized.

To ensure fair borrowing *least recently degraded/ upgraded* (LRDU) policy is adopted with the above mentioned scheme. According to this policy, among the ongoing streams of same type having the same level of quality one that has been degraded least recently is borrowed from first. Conversely, among the ongoing streams of same type having the same level of quality one that has been upgraded least recently is restored first. The property that multicast streams are used by many users can be exploited to further increase the average video quality received by each user adopting LRDU *patch only* (LRDUPO) policy with the above mentioned scheme which borrows bandwidth only from patch streams. The detailed algorithms for these borrowing schemes are not presented in this paper due to the lack of space.

4. Theoretical analysis

Consider a video of length L. If server capacity is infinite then controlled multicast scheme can be modeled as a $M/G/\infty$ queuing system where the arrival rate is a Poisson distribution with mean λ and the service time is a general distribution with mean $1/\mu$. When the threshold is T, a new regular stream starts once every $T+1/\lambda$ time. The total number of arrivals in this interval is $\lambda T+1$. It can be assumed that the first arrival among them is served by a regular stream and the others are served by a new patch stream. So the average service time is given by:-

$$\bar{x}=\frac{1}{\mu}=\frac{1}{\lambda T+1}\cdot L+\frac{\lambda T}{\lambda T+1}\cdot\frac{T}{2}=\frac{L+\lambda T^2/2}{\lambda T+1} \quad (1)$$

and the traffic intensity by:-

$$\rho=\lambda\bar{x}=\frac{\lambda(L+\lambda T^2/2)}{\lambda T+1}. \quad (2)$$

The minimum traffic intensity can be calculated as

$$\rho_{\min}=\sqrt{2L\lambda+1}-1 \quad (3)$$

when

$$T=\left(\sqrt{2L\lambda+1}-1\right)/\lambda. \quad (4)$$

An alternate derivation can be found in [5].

Now consider a finite bandwidth video server that can deliver m video streams of bandwidth $B_{\max}$ to serve the user requests and user requests are blocked whenever insufficient bandwidth is available. In this case, controlled multicast can be modeled as a $M/G/m/m$ queuing system with mean arrival rate λ and mean service time $1/\mu$. The successful arrival rate $\lambda_a = \lambda(1-P_B^m)$, where P_B^m is the quiescent state probability. So the mean service time $\bar{x} = 1/\mu$ can be obtained from (1) by substituting λ with λ_a and the traffic intensity can be calculated as $\rho = \lambda\bar{x}$. The blocking probability is therefore given by:-

$$P_B^m = \frac{\rho^m}{m!} \Big/ \sum_{j=0}^{m} \frac{\rho^j}{j!} \tag{5}$$

which can be expressed as $P_B^m - f(P_B^m) = 0$. Solving (5) numerically using the *Newton-Raphson* method P_B^m can be determined for any arbitrary value of T. The expected bandwidth allocated by the server to all the ongoing streams is thus:-

$$E[BW_m] = \sum_{k=0}^{m} kB_{\max}P_k = \left(1-P_B^m\right)\rho B_{\max} \tag{6}$$

and bandwidth utilization can be calculated as

$$U_m = \frac{E[BW_m]}{m.B_{\max}} = \frac{\left(1-P_B^m\right)\rho}{m}. \tag{7}$$

The expected amount of bandwidth allocated to each stream is $E[S_m] = B_{\max}$.

Assume that in case of a bandwidth crisis, any amount of bandwidth can be borrowed from ongoing streams and allocated to new streams. The minimum bandwidth needed for each stream is $B_{\min} < B_{\max}$ so the number of streams is bounded by:-

$$n = \left\lfloor \frac{m \times B_{\max}}{B_{\min}} \right\rfloor \geq m. \tag{8}$$

The blocking probability for the borrowing scheme P_B^n can be calculated by substituting m by n in (5). Obviously, $P_B^n \leq P_B^m$ for $n \geq m$ since $\rho^n/n!$ grows slower than $\sum_{j=0}^{n} \rho^n/n!$. If the number of ongoing streams $k \leq m$ then a total $kB_{\max}$ bandwidth is allocated where every stream uses $B_{\max}$ bandwidth, otherwise $(i.e.\ m < k \leq n)$ total $mB_{\max}$ bandwidth is allocated where every stream uses on the average $(m/k)B_{\max}$ bandwidth. The expected amount of bandwidth allocated the by the server to all the ongoing streams is

$$E[BW_n] = \sum_{k=0}^{m} kB_{\max}P_k + \sum_{k=m+1}^{n} mB_{\max}P_k. \tag{9}$$

Therefore bandwidth utilization is

$$U_n = \frac{E[BW_n]}{m.B_{\max}}. \tag{10}$$

The expected amount of bandwidth allocated to each stream is given by

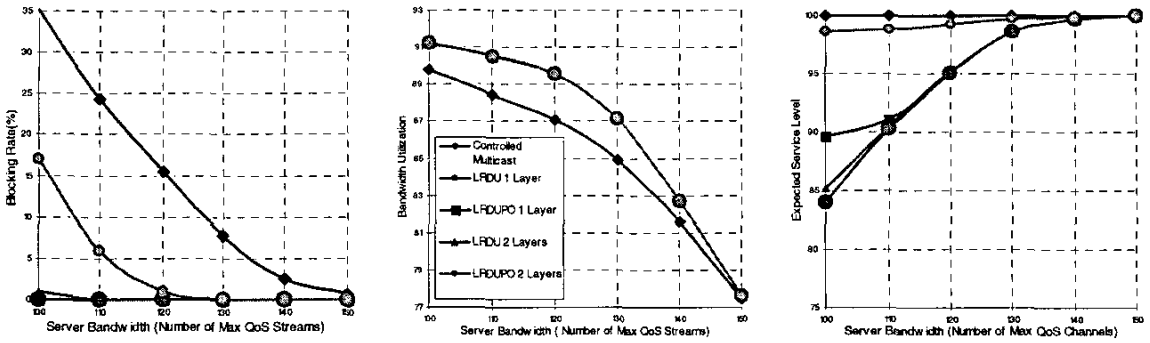
$$E[S_n] = \sum_{k=0}^{m} p_k.B_{\max} + \sum_{k=m+1}^{n} p_k.\frac{m \times B_{\max}}{k} \leq E[S_m]. \tag{11}$$

5. Simulation results

The proposed LRDU and LRDUPO bandwidth borrowing approaches are compared with the controlled multicast scheme. It is assumed that a user reneges without waiting if sufficient bandwidth is not available to deliver a new stream with minimum QoS. Videos are encoded with 4 layers, --1st, 2nd, 3rd and 4th requiring $4b$, $3b$, $2b$ and b units of bandwidth respectively, where the 1st layer is the base layer and the rest are successive enhancement layers. The user arrival rate is assumed to be a Poisson distribution with a mean of 10/minute. There are 20 movies each of length 120 minutes. The videos are selected by users according to Zipf distribution with skew factor 0.271 i.e. the probability of choosing video i is $p_i = f_i / \sum_{k=1}^{N} f_k$ where $f_i = 1/i^{1-\theta}$, N is the total number of videos and θ is the skew factor. The threshold value calculated in (4) is used to start a new regular stream. The user blocking rate, utilization of bandwidth, average service level is calculated. The service level is defined as the amount of bandwidth a user is served with as a fraction of bandwidth required for a maximum QoS video stream. Each simulation runs for 10,000 user requests. The suffix '1 Layer'/'2 Layers' in the legends of Fig. 1 indicates that at least 1 layer/2 layers must be delivered to ensure the minimum QoS to video streams.

Fig. 1(a) shows the blocking rate (%) vs. server bandwidth (number of maximum QoS streams), for all three schemes – controlled multicast, LRDU, LRDUPO. In controlled multicast scheme the blocking rate is very high when server bandwidth is lower, but blocking rate decreases significantly as server bandwidth increases. The two new borrowing schemes LRDU and LRDUPO always have lower blocking rate than controlled multicast since they allocate bandwidth for new stream by borrowing when there is shortage. Since LRDUPO has less amount of borrow-able bandwidth, it has slightly higher blocking rate than LRDU.

Fig. 1(b) shows bandwidth utilization (%) vs. server bandwidth (number of maximum QoS streams) performance of the three schemes. Bandwidth utilization for LRDU and LRDU patch only are almost equal. Bandwidth utilization for controlled multicast



(a) (b) (c)

Fig. 1: Server bandwidth (number of maximum QoS streams) vs. (a) blocking rate (%); (b) bandwidth utilization (%); (c) expected service level (%).

scheme is lower than the two borrowing schemes due to higher blocking rate.

Fig. 1(c) shows expected service level (%) vs. server bandwidth (number of maximum QoS streams) for the three schemes. Controlled multicast scheme always has 100% expected service level because this scheme does not allow bandwidth borrowing. The two borrowing schemes have expected service level lower than the controlled multicast scheme. When the server bandwidth is less, LRDUPO has a higher expected service level than LRDU but when server bandwidth is higher their service levels are very close. This is because less amount of bandwidth is borrowed when the server has sufficient free bandwidth.

6. Conclusion

The proposed bandwidth borrowing schemes – LRDU and LRDUPO reduces user request blocking rate and increases bandwidth utilization by degrading video quality temporarily in a fair manner. In terms of overall performance LRDUPO is better since this provides significantly higher video quality by blocking slightly more user requests than LRDU.

7. References

[1] A. Dan, P. Shahabuddin, and D. Sitaram, "Scheduling policies for an on-demand video server with batching," in *Proc. of ACM Multimedia*, pp. 168-179, October 1994.

[2] S. W. Carter, and D. D. E. Long, "Improving video-on-demand server efficiency through stream tapping," in *Proc. of the Sixth Int'l Conference on Computer Comm. and Networks (ICCCN' 97)*, pp. 15-23, October 1997.

[3] K. Hua, Y. Cai, and S. Sheu, "Patching: A multicast technique for true video-on-demand services," in *Proc. of ACM Multimedia*, September 1998.

[4] S. Sen, L. Gao, J. Rexford, and D. Towsley, "Optimal patching schemes for efficient multimedia streaming," in *Proc. of Ninth International Workshop on Network and Operating Systems Support for digital Audio and Video (NOSSAV' 99)*, June 1999.

[5] L. Gao, and D. Twosly, "Supplying instantaneous video-on-demand services using controlled multicast," in *Proc. of IEEE International Conference on Multimedia Computing and Systems*, 1999.

[6] L. Golubchik, J. Lui, and R. Muntz, "Adaptive piggybacking: A novel technique for data sharing in video-on-demand storage servers," *ACM Multimedia Systems Journal*, vol. 4, no. 3, 1996.

[7] C. Aggarwal, J. Wolf, and P. S. Yu, "On optimal piggyback merging policies for video-on-demand systems," in *Proc.of ACM SIGMETRICS Conference*, May 1996.

[8] S. Viswanathan, and T. Imielienski, "Metropolitan area video-on-demand service using pyramid broadcasting," *Multimedia Systems*, vol. 4, pp. 197-208, August 1996.

[9] K. A. Hua, and S. Sheu, "Skyscraper broadcasting: a new broadcasting scheme for metropolitan video-on-demand systems," *in Proc. of SIGGCOMM 97 Conference*, pp. 89-100, September 1997.

[10] L. Jhun, and L. Tseng, "Harmonic broadcasting protocols for video-on-demand service," *IEEE transactions on broadcasting*, vol. 43, pp. 268-271, September 1997.

[11] J. -F. Paris, S. W. Carter, and D. D. E. Long, "Efficient broadcasting protocols for video on demand," in *Proc. of the 6th International Symposium on Modelling, Analysis, and Simulation of Computing and Telecom Systems*, pp. 127-132, July 1998.

[12] J. -F. Paris, S. W. Carter, and D. D. E. Long, "A hybrid broadcasting protocol for video-on-demand systems," in *Proc. of the 1999 Multimedia and Networking Conference*, pp. 317-326, January 1999.

[13] M. -H. Yang, C. -H. Chang, and Y. -C. Tseng, "A borrow-and-return model to reduce client waiting time for broadcasting-based VOD services," *IEEE transactions on broadcasting*, vol. 49, no. 2, pp. 162-169, June 2003.