Scalable Delivery and Pricing of Streaming Media with Advertisements*

Musab Al-Hadrusi
hadrusi@wayne.edu

Nabil J. Sarhan
nabil@ece.eng.wayne.edu

Department of Electrical and Computer Engineering
Wayne State University
Detroit, MI 48202

ABSTRACT

This paper presents a delivery framework for streaming media with advertisements and an associated pricing model. The delivery model combines the benefits of periodic broadcasting and stream merging. The advertisements' revenues are used to subsidize the price of the media content. The pricing is determined based on the total ads' viewing time. Moreover, this paper presents three modified scheduling policies that are well suited to the proposed delivery framework and analyzes their effectiveness through simulation.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols; C.4 [**Computer Systems Organization**]: Performance of Systems; I.6.5 [**Simulation and Modeling**]: Model Development

General Terms

Design, Economics, Performance

Keywords

Media streaming, periodic broadcasting, pricing, scheduling, stream merging.

1. INTRODUCTION

The interest in media streaming has grown dramatically. Unfortunately, the distribution of streaming media faces a significant scalability challenge because of the high server and network requirements. Therefore, numerous techniques have been proposed to deal with this challenge, especially in the areas of *media delivery* (also called *resource sharing*) and *request scheduling*.

Scalable media delivery can be done in a *client-pull* or a *server-push* fashion, depending on whether the channels are allocated on demand or reserved in advance, respectively. The first category includes *stream merging* techniques [8, 5, 7, 16, 12, 15], which reduce the delivery cost by aggregating clients into larger groups that share the same multicast streams. *Patching* is a simple yet efficient stream merging technique. With Patching, a new request joins immediately the latest multicast stream for the object and receives the missing portion as a patch using a unicast stream. When the playback of the patch is completed, the client continues the playback of the remaining portion using the data received from the multicast stream and already buffered locally. The degrees of resource sharing achieved by client-pull media delivery techniques depend greatly on how the waiting requests are scheduled for service. Server-push techniques, also called *Periodic Broadcasting* techniques [9, 11, 13, 10, 18] divide each media file into multiple segments and broadcast each segment periodically on dedicated server channels. They are cost-performance effective for highly popular content but lead to channel underutilization when the request arrival rate is not sufficiently high.

The overwhelming majority of prior studies on media delivery focused on regular content without any media advertisements. The use of ads is important for many reasons, including the following.

- Ads generate revenue, which can be used to pay for the service cost, generate profit, or subsidize the paid content.
- A streaming solution that supports ads can essentially convert passive startup waiting times for service to active waiting times (i.e., watching ads while waiting for the playback of the desired media content). Today, even for short videos with medium quality, users of online video websites may experience significant delays. The transition, in the near future, to streaming long videos (such as full-length movies) at high quality may lead to even longer delays.
- Many users like to watch some types of ads, such as movie trailers, to know about other interesting movies to watch.
- Using ads results in better aggregation of the requests and thus can reduce the delivery costs.

This paper presents a framework for scalable delivery of media content with advertisements including a pricing model. The delivery framework combines the benefits of stream merging and periodic broadcasting. A client starts by joining an ads' broadcast channel for some time and then receives the requested media by stream merging. The revenues generated from the ads are used to subsidize the price. Subsidizing the price helps attract more clients, thereby increasing the overall revenue [3]. Pricing depends on the experienced quality-of-service (QoS). In particular, clients with larger ads' viewing time get lower prices. Since the price has to be set before the client makes a selection of the media content to playback, waiting-time prediction [2] can be used to estimate the ads' viewing period based on the requested media and the system's current state.

*This work is supported in part by NSF grant CNS-0626861.

MM'07, September 23–28, 2007, Augsburg, Bavaria, Germany.

In addition, this paper presents three modified scheduling policies for the proposed delivery framework and analyzes their effectiveness in terms of customer defection (i.e., turn-away) probability, ads' viewing time, and average price.

The rest of the paper is organized as follows. Section 2 presents the proposed delivery framework and pricing model. Section 3 presents the proposed modifications of scheduling policies. Section 4 discusses the performance evaluation methodology and main results. Subsequently, Section 5 discusses the related work. Finally, conclusions are drawn.

2. PROPOSED DELIVERY FRAMEWORK AND PRICING MODEL

The proposed delivery framework combines the benefits of stream merging and periodic broadcasting. Periodic broadcasting is used for the ads because they are potentially accessed more frequently than the individual primary media content, especially if each client is required to view a minimum number of ads. The primary media contents, however, are delivered using a scalable stream merging technique and can benefit from a high degree of aggregation because of the use of ads.

The main characteristics of the framework can be summarized as follows.

- Clients start by joining an ads broadcast channel for some time and then receive the requested media by stream merging.
- Ads are combined and broadcast on dedicated server channels. Hence, when beginning listening to an ads' channel, the client views different ads until streaming of the desired media commences. Multiple channels can be used with time-shifted versions of the combined ads, as shown in Figure 1, to reduce the waiting time for reaching the beginning of an ad. In the figure, a total of four ads are supported and broadcast on three channels. The ad interval is 30 seconds. With these three channels, the maximum time for a new request to reach the beginning of an ad is $30/3 = 10$ seconds, and the average time is $10/2 = 5$ seconds. With only one channel, the maximum waiting time to reach an ad is 30 seconds. The ads can be of varying durations, but a smart allocation scheme is required.
- Ads are only viewed prior to watching the actual media content. Uninterrupted viewing of the primary media allows for a more enjoyable playback experience.

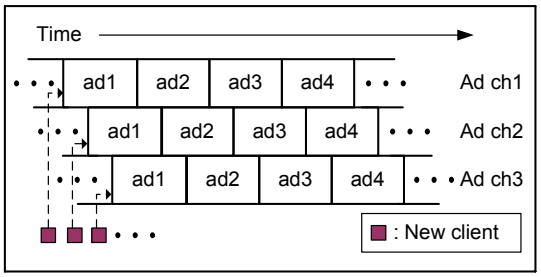


Figure 1: Ads' Broadcast Channels

In the pricing model, the price for streaming a media file is determined based on the client's total ads' viewing time. The system should estimate the expected ads' viewing time for each potential request dynamically because the clients need to know the price before purchasing the service. This can be done using waiting-time prediction [2] based on the current state of the system and the requested media file. Hence, the system provides clients with updated menus containing the list of media files and the corresponding expected ads' viewing durations and prices. The revenues generated from the ads are used to subsidize the price. A certain portion can be distributed uniformly among all clients and the rest distributed proportionally to the client total ads' viewing time. Price subsidization attracts more clients and thus increases the overall revenue.

3. PROPOSED SCHEDULING MODIFICATIONS

A video streaming server maintains a waiting queue for every video and applies a scheduling policy to select an appropriate queue for service whenever it has an available *channel*. A channel is a set of resources (network bandwidth, disk I/O bandwidth, etc.) needed to deliver a multimedia stream. All requests in the selected queue can be serviced using only one channel. The number of channels is referred to as *server capacity*.

The main scheduling policies include *First Come First Serve* (FCFS) [6], *Maximum Queue Length* (MQL) [6], *Maximum Factored Queue Length* (MFQL) [1], and *Minimum Cost First* (MCF) [17]. MCF achieves the best overall performance by capturing the significant variation in stream lengths caused by stream merging techniques through selecting the requests requiring the least cost. The length of the stream (in time) is directly proportional to the cost of servicing that stream since the server allocates a channel for the entire time the stream is active. *MCF-P (RAP)* is the preferred implementation of MCF. It selects the queue with the least cost per request and treats regular streams and transition patches in a preferential manner because they are shared by later patches.

Most existing scheduling policies are not well suited for the proposed delivery framework. For example, MCF, MQL, and MFQL attempt to serve requests as soon as possible, thereby reducing significantly the ads' viewing time. Thus, we present two modifications of MCF-P (RAP) to ensure that ads are viewed by a large number of users: *Each n* and *Any n*. *Each n* considers a video for scheduling only if each waiting request for it has viewed at least n ads, whereas *Any n* considers a video for service only if any one of its waiting requests has viewed at least n ads. We compare the effectiveness of MAT, MCF (Any n), and MCF (Each n). Here, n is set to 2 since it provides the best overall performance with the analyzed workload. These modified versions can work with MQL and MFQL but we primarily use them for MCF-P (RAP) because it is the best performer.

Moreover, we propose a policy, called *Maximum Ads Time* (MAT), which selects for service the video whose waiting requests have the longest total ads' viewing time.

4. EVALUATION METHODOLOGY AND RESULTS

We have extended the validated media streaming simulator in [14] by implementing the proposed delivery framework and pricing model. Next, the workload characteristics and performance metrics are discussed.

4.1 Workload Characteristics

Table 1 summarizes the workload characteristics used. Like most prior studies, we assume that the arrival of the requests to the server follows a Poisson Process with an average arrival rate λ and that the access to videos is highly localized and follows a Zipf-like distribution with skewness parameter $\theta = 0.271$. We characterize the waiting tolerance of customers in terms of the number of ads by a Poisson distribution. Combining this study with waiting-time pre-

diction is left for a later study. Waiting-time prediction is likely to impact the waiting tolerance model.

We consider here a commercial *Movie-on-Demand* system with 120 titles, each of which is 120-minute long. Stream merging is done using Patching. In the future, we will evaluate other hierarchical merging techniques, such as *Earliest Reachable Merge Target* (ERMT) [7]. We analyze the impacts of both server capacity (i.e., number of server channels) and arrival rate. Without loss of generality, we assume here a *cost-plus* model for the price. The price covers the movie royalty fee, delivery fee, and operational cost minus subsidization credit. All revenues from the ads are distributed to the clients proportionally to their total ads' viewing times. In the discussed example, the revenue per ad per user is 10 cents, the movie royalty fee is 70 cents, and the delivery cost per GB is 50 cents. Based on service positioning analysis, the service provider seeks to get 70 cents per movie request to cover their operational cost and attain the sought profit.

Table 1: Summary of Workload Characteristics

Parameter	**Model/Value(s)**
Request Arrival	Poisson Process
Request Arrival Rate	Variable, Default = 40 Req./min
Server Capacity	Variable, Default = 600 channels
Video Access	Zipf-Like Skewness Parameter $\theta = 0.271$
Number of Movies	120
Movie Length	120 min
Waiting Tolerance Model	Poisson, min = 3 ads, mean= 5 ads, max = 8 ads

4.2 Performance Metrics

The analyzed performance metrics here are *customer defection probability*, *average number of ads viewed per client*, and *price*. The first can be defined as the probability that a customer leaves without being serviced as a result of a waiting time exceeding its tolerance.

The overall revenue is challenging to estimate, although it can be given simply as the product of the volume sold and the price. The volume here is the total number of streams delivered to clients, which directly depends on the customer defection probability. The complication happens because the price influences the arrival rate and number of streams delivered. Thus, subsidizing the price can attract more clients and can eventually increase the overall revenue. By increasing the arrival rate, the delivery costs also decrease because of the higher degrees of request aggregation and stream merging. Determining the overall impact of scheduling policy on the revenue (and profit) requires knowledge of the function of arrival rate versus price and will be investigated in future work.

4.3 Result Presentation and Analysis

Figure 2 compares the effectiveness of the three scheduling policies: MCF (Any 2), MCF (Each 2), and MAT. MCF (Any 2) performs significantly better than both MCF (Each 2) and MAT in customer defection rate (which directly translates to throughput). As expected, MCF (Each 2) increases the number of ads and thus reduces the price more than MCF (Any 2). MAT tends to perform generally between MCF (Each 2) and MCF (Any 2) in the three metrics.

Figure 3 shows the percentage reduction in the price by subsidization for each policy. These results suggest that the ads can reduce the price by 7% to 45%. The price reduction decreases with the server capacity because the number of viewed ads decreases. The impact of server capacity on price reduction is most significant with MAT because it does not force any minimum ads' viewing requirement.

The impact of arrival rate is illustrated in Figures 4 and 5 on defection probability and price, respectively. The relative behavior of the three policies remains essentially the same.

Although MCF (Each 2) and MAT reduce the price more than MCF (Any 2) by increasing the ads' viewing time, MCF (Any 2) is expected to generate higher profit and revenue because the increase in throughput will have a more significant impact than the increase in the arrival rate (as a result of price reduction).

5. RELATED WORK

Supporting ads has been discussed in only few studies. In [4], the primary data and ads are delivered using piggybacking on the same channels. Piggybacking adjusts the movie playback rate so that two streams can merge with each other, thereby impacting the playback quality and suffering from technical challenges and complexities to change the movie playback rate. Moreover, ads are inserted randomly multiple times during the playback of the primary media. The study [3] provided a general discussion of pricing based on the techniques in [4]. It discussed the general relationships among arrival rate, price, and ads' ratio (to the total user viewing time). It also discussed a pricing model similar to the one in this paper but without waiting-time prediction.

6. CONCLUSIONS

In this paper, we have presented and analyzed a delivery framework for streaming media with advertisements and an associated pricing model. The delivery model combines the benefits of periodic broadcasting and stream merging. The advertisements' revenues are used to subsidize the price of the media streaming. The pricing is determined based on the total ads' viewing time. Moreover, we have presented three modified scheduling policies for the proposed framework and have compared their effectiveness. These policies are *Maximum Ads Time* and two variants of *Minimum Cost First* (MCF): *Any n* and *Each n*. The analyzed metrics include customer defection probability, average ads' viewing time per client, and the price. The preliminary results indicate that MCF (Any n) achieves the best overall performance. Moreover, it can control the average ads' viewing time by adjusting n.

7. REFERENCES

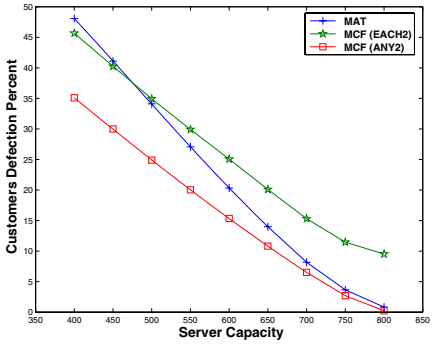
[1] C. C. Aggarwal, J. L. Wolf, and P. S. Yu. The maximum factor queue length batching scheme for Video-on-Demand systems. *IEEE Trans. on Computers*, 50(2):97–110, Feb. 2001.

[2] M. Alsmirat, M. Al-Hadrusi, and N. J. Sarhan. Analysis of waiting-time predictability in scalable media streaming. In *Proceedings of ACM Multimedia (To Appear)*, Sept. 2007.

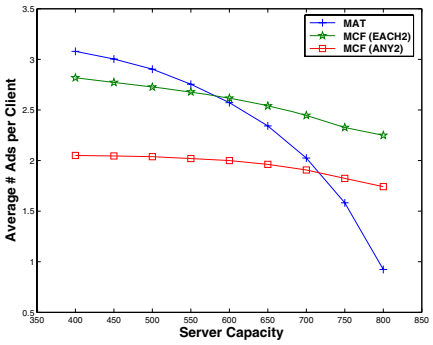
[3] P. Basu and T. D. C. Little. Pricing considerations in video-on-demand systems. In *Proceedings of ACM Multimedia*, pages 359–361, Nov. 2000.

[4] P. Basu, A. Narayanan, W. Ke, T. D. C. Little, and A. Bestavros. Optimal scheduling of secondary content for aggregation in video-on-demand systems. In *Proceedings of International Conference on Computer Communications and Networks*, pages 104–109, Oct. 1999.

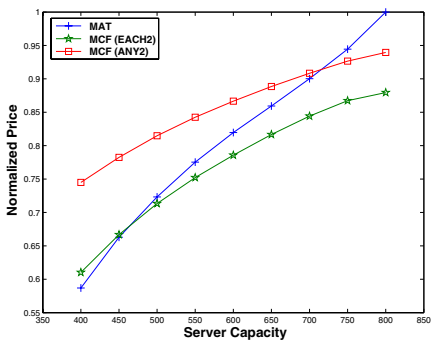
[5] Y. Cai and K. A. Hua. An efficient bandwidth-sharing technique for true video on demand systems. In *Proc. of ACM Multimedia*, pages 211–214, Oct. 1999.



(a) Customer Defection Probability



(b) Number of Viewed Ads



(c) Normalized Price

Figure 2: Effectiveness of Scheduling Policies

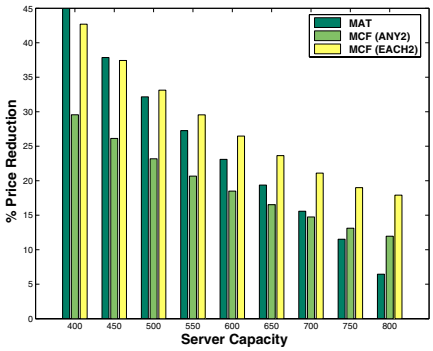


Figure 3: Analysis of Price Reduction

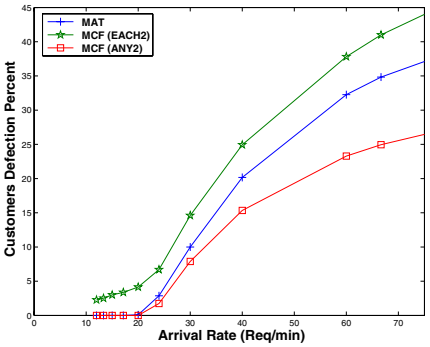


Figure 4: Impact of Arrival Rate on Customer Defection Probability

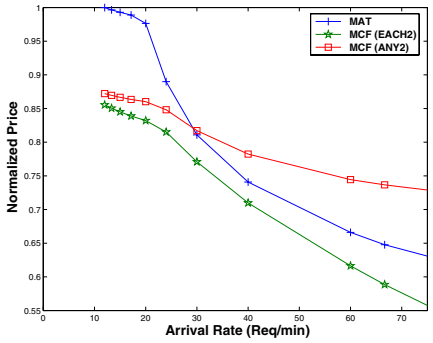


Figure 5: Impact of Arrival Rate on Price

[6] A. Dan, D. Sitaram, and P. Shahabuddin. Scheduling policies for an on-demand video server with batching. In *Proc. of ACM Multimedia*, pages 391–398, Oct. 1994.

[7] D. L. Eager, M. K. Vernon, and J. Zahorjan. Optimal and efficient merging schedules for Video-on-Demand servers. In *Proc. of ACM Multimedia*, pages 199–202, Oct. 1999.

[8] K. A. Hua, Y. Cai, and S. Sheu. Patching: A multicast technique for true Video-on-Demand services. In *Proc. of ACM Multimedia*, pages 191–200, 1998.

[9] K. A. Hua and S. Sheu. Skyscraper broadcasting: A new broadcasting scheme for metropolitan Video-on-Demand system. In *Proc. of ACM SIGCOMM*, pages 89–100, Sept. 1997.

[10] C. Huang, R. Janakiraman, and L. Xu. Loss-resilient on-demand media streaming using priority encoding. In *Proc. of ACM Multimedia*, pages 152–159, Oct. 2004.

[11] L. Juhn and L. Tseng. Harmonic broadcasting for Video-on-Demand service. *IEEE Trans. on Broadcasting*, 43(3):268–271, Sept. 1997.

[12] H. Ma, G. K. Shin, and W. Wu. Best-effort patching for multicast true VoD service. *Multimedia Tools Appl.*, 26(1):101–122, 2005.

[13] J.-F. Pâris, S. W. Carter, and D. D. E. Long. Efficient broadcasting protocols for video on demand. In *Proc. of the Int'l Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 127–132, July 1998.

[14] B. Qudah and N. J. Sarhan. Analysis of resource sharing and cache management techniques in scalable video-on-demand. In *Proc. of the 14th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 327–334, Sept. 2006.

[15] B. Qudah and N. J. Sarhan. Towards scalable delivery of video streams to heterogeneous receivers. In *Proc. of ACM Multimedia*, pages 347–356, Oct. 2006.

[16] M. Rocha, M. Maia, I. Cunha, J. Almeida, and S. Campos. Scalable media streaming to interactive users. In *Proc. of ACM Multimedia*, pages 966–975, Nov. 2005.

[17] N. J. Sarhan and B. Qudah. Efficient cost-based scheduling for scalable media streaming. In *Proc. of Multimedia Computing and Networking Conf. (MMCN)*, January 2007.

[18] L. Shi, P. Sessini, A. Mahanti, Z. Li, and D. L. Eager. Scalable streaming for heterogeneous clients. In *Proceedings of ACM Multimedia*, pages 337–346, Oct. 2006.