On Achieving Profit in Providing Near Video-on-Demand Services

S.-H. Gary Chan Fouad A. Tobagi
Department of Electrical Engineering
Stanford University
Stanford, CA 94305

Abstract— **In near video-on-demand, requests for the same movie arriving within a period of time are grouped together (i.e., batched) and served with a single multicast stream; thereby reducing the bandwidth requirement compared with the unbatched case. We consider here that the use of a multicast channel comes with a cost. Since delayed users may cancel their requests while being batched, this uncollected revenue has to be appropriately balanced with the channel usage cost in order to achieve maximum profit. We address here profit issues for a number of batching schemes. In the window-based schemes, users are batched for a fixed period of time before they are served; in the batch-size based scheme, users are served once a certain number of them are collected. We study the system profit in terms of user reneging behavior, movie popularity, and multicast channel cost. By combining the advantages of the window-based and the batch-size based schemes, we introduce a scheme which adapts to the fluctuating request rate to achieve high profit.**

Keywords— **Near video-on-demand, request batching and multicasting, network channel cost, user reneging behavior, profit issues**

I. Introduction

Video-on-demand (VOD) refers to video services in which a user is able to request from a server any video content at any time. VOD encompasses many applications such as movie-on-demand, news-on-demand, distance learning, home shopping, various interactive training programs, etc.[1], [2], [3] In pure-VOD, each user is assigned its own dedicated unicast stream; hence it enjoys great flexibility in interacting with its displaying video. However, pure-VOD becomes expensive when a large number of concurrent requests have to be accommodated.

When the video content is popular, the use of a single multicast stream serving many users simultaneously becomes more cost-effective. This is accomplished by grouping (i.e., batching) many requests for a given content and serving them with a single multicast stream. This is referred to as near-VOD [4]–[18]. However, the bandwidth saving as compared to pure-VOD is achieved at the expense of user start-up delay. Since each user no longer has a dedicated stream, such technique is effective when user interactivity does not have to be flexible or is not essential, as is the case with such applications as movie-on-demand. (Providing interactivity in the multicast environment has been discussed in [17], [18].) Today, request batching is practically used in many satellite and cable network-based movie-on-demand services.

A near video-on-demand system consists of the following three components (Ref. Fig. 1) :
• Video servers — The video servers store a number of movies characterized by their duration, popularity and streaming rate. They have finite streaming capacities which may be partitioned or shared among the movies [19], [20], [21], [22]. In a near-VOD system, the main issue is to appropriately assign the limited streaming capacity to various requests by means of batching.

This work was supported in part by a grant from the Center for Telecommunications at Stanford University.

S.-H. Gary Chan was with Stanford University when this work was done. He is currently with the Department of Computer Science, University of California, Davis, CA 95616.

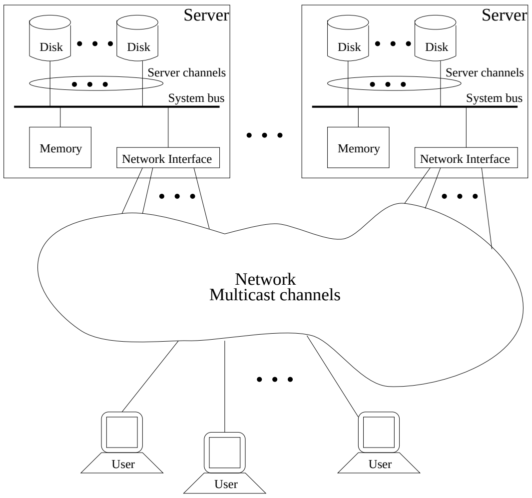

Fig. 1. A near video-on-demand system.

• Network — The network offers the multicast channels needed. A certain number of channels may be leased by the service provider and always available (already paid for). On the other hand, multicast channels may be requested on-demand at some cost (e.g., satellite channels), in which case the service provider would be interested in amortizing the cost of the channels to guarantee a certain level of profit.
• Users — The users make requests to view certain movies. These requests are characterized by the stochastic process representing the arrival in time of a request, which depends on factors such as the time of the day, the occurrence of some events, etc., and the movie's popularity (the probability of selecting the movie). What is important for the users is the waiting time from when the user places a request until the movie starts being displayed. Depending on the waiting time incurred, a user may cancel its request (i.e., renege). Since a frustrated user may never visit the system again, the user loss rate should be kept low. The reneging behavior of the users is an important consideration in the design of a near VOD system. Many user reneging models have been considered in the literature, and we will study a number of them here.

We briefly describe previous work as follows. Gelman and Halfin consider a video system in which the server channels are partitioned among the movies, and analyze a number of batching schemes in terms of the user loss rate [4]. Asit Dan et al. consider in [5] that the server channels are shared among the movies and study the user loss rate and delay issues. Tsiolis and Vernon consider in [14] that groups of movies share a number of channels. Aggarwal et al. in [6] propose a batching scheme which weights the number of waiting users for a movie by a factor biased against its popularity, so as to achieve uniformity in the user loss rate across the movies. In [7], Almeroth et al. examine the adaptability with respect to the request rate of a few batching schemes in terms of their user loss rates and average user delay. In [8], the minimization of user loss rate given a certain number of server channels and user's reneging behavior is studied.

All the above work concentrated on the streaming capacity of the servers as being given and already paid for, and therefore addressed the issue of how to assign the available channels to requests so as to achieve a certain objective in the loss rate. We consider here that the network channels can be acquired on-demand with a cost incurred each time they are used. Under this circumstance, the loss rate may no longer be the only measure of performance. From the service provider's point of view, profit is an important consideration; and the way of batching requests so that the channel usage cost is amortized while offering an acceptable loss rate becomes an important design goal. We analyze a number of batching schemes in this light given user reneging behavior. We primarily consider here the case in which the number of on-demand channels is unlimited. Accordingly, the servicing of requests pertaining to a given movie is independent of the servicing of requests for other movies, and hence it is sufficient to consider the single movie case. On-demand network channels have been studied in [15], in which the performance of a number of batching schemes is studied under the condition that user reneging is negligible; we introduce user reneging here. The case in which the number of available channels for a given movie is limited has been addressed in [16]. More advanced batching schemes were reported in [9]–[12][17]. Our work can be combined with theirs to achieve higher profit and lower user loss rate.

We consider maximizing the profit rate which translates to the profit achieved over a long period of time. System profit depends on the length of the batching period: if the period is too short, many network channels would be used with only a few users per channel, leading to high usage cost. As the period increases, the batch size increases, but due to user reneging, the batch size reaches a limit; thus if the batching period gets too long, the profit over a period of time decreases. We find the parameters to achieve maximum or high profit for two well-known basic batching schemes, namely, the window-size based schemes and the batch-size based scheme. In the window-size based schemes, a maximum user delay is guaranteed. With the batch-size based scheme, per stream revenue (and thus per stream profit) is guaranteed. We then introduce a new scheme which combines the key advantages of the window-size based schemes (namely high profit when arrival rate is high) and of the batch-size based scheme (namely guaranteed per stream profit when the arrival rate drops).

This paper is organized as follows. We first describe the batching schemes in Sect. II. In Sect. III, we present the profit measures and analyze the schemes with loss-independent arrival rate. In Sect. IV, numerical examples and comparisons are provided. We discuss in Sect. V how our results can be extended to loss-dependent arrival rate. We conclude in Sect. VI.

II. Batching Schemes

In this section, we describe the two basic batching schemes, and then introduce the combined scheme.

A. Window-size based schemes

Consider a movie of duration T_h minutes, which is scheduled to be shown once every exactly W minutes (the batching window). In *fixed-gating*, a user first makes reservation to view the movie, and would be scheduled to be served at the upcoming movie show-time following the request. A user may cancel its request while waiting. A movie showing is omitted if there is no outstanding request for it by the end of its corresponding batching window.

Another batching scheme which also guarantees a delay bound of W minutes is the one in which the first user of a batch is forced to wait W minutes (the batching window) before it is served. The scheme is referred to as *auto-gated scheduling.* Note that since the first user in a batch may renege (in which case the batching window is advanced to the next request), the show-time of the movie can no longer be advertised to the users. Obviously, W must not exceed the maximum waiting tolerance of all the users in the system; otherwise, a batch could never be formed.

B. Batch-size based scheme

In general, the concern of a service provider is to have as many users in a batch as possible, since the system profit is related to the average batch size. Let P_V dollars be the pay-per-view for a movie, with the total cost of using a channel as the sum of a fixed cost C dollars and a variable cost proportional to the number of users in the batch (the proportionality constant being α dollars/user). If $\overline{N}$ is the average number of users in a batch, the average per-batch profit is given by $\overline{N}P_V - (C+\alpha\overline{N})$. Defining $P \triangleq P_V - \alpha$ (the net revenue per user), a batching scheme is hence profitable if $\overline{N} \geq C/P \triangleq K$, i.e., the average batch size is no less than a number K. In the batch-size based scheme, we secure the profitability by batching $M \geq K$ users together before multicasting the movie, regardless of how long it takes to collect the M users. Note that since users experience uncertain waiting time, the time at which the movie is displayed cannot be advertised to the users.

C. Combined scheme

Note that we may optimize the batching parameters (i.e., W in the window-based schemes and M in the batch-size based scheme) for a movie to achieve maximum profit with a target request rate λ_0 req./min. in mind. In reality, the request rate may fluctuate around λ_0. In the auto-gated scheduling, the average batch size increases with the arrival rate, and hence the profit increases quite fast when the arrival rate increases beyond λ_0. However, since the scheme cannot guarantee every batch to be profitable, when the arrival rate drops to a low value, the system profit suffers greatly. On the other hand, the batch-size based scheme guarantees that each batch is profitable no matter what the arrival rate is. However, since the batch size is fixed, the profit does not increase fast enough when the arrival rate gets higher than λ_0.

Therefore we propose a combined scheme which operates according to the window-based auto-gated scheduling when the arrival rate gets high, and according to the batch-size based scheme when the rate drops (Such scheme is very similar to the adaptive scheme given in [15], except that we have taken $W_{max} = \infty$, and consider user reneging here): This scheme keeps batching requests until the batch size is no less than a parameter $M \geq K$ (so as to ensure profit) *and* the batching period (the time between the first request in the batch and the movie show-time) is no less than W minutes (so as to safeguard against too short a batching period).

III. Profit Analysis

In this section, we consider how profit can be maximized with the aforementioned batching schemes. We are interested in the following closely-related performance measures (the user delay distribution has been analyzed in [23]): i) the average per-batch profit, given by $(\overline{N}-K)P$; ii) the profit rate θ (dollars/minute), given by $(\overline{N}-K)P/\overline{T}$, where $\overline{T}$ minutes is the average time between successive channel assignments; further define $\hat{\theta}$ as the normalized profit rate with respect to P, i.e., $\hat{\theta} \triangleq \theta/P$; and iii) the system throughput λ' (the number of accepted requests served per minute), given by $\lambda' = \overline{N}/\overline{T}$. The loss rate p_L is hence given by $p_L = 1-\lambda'/\lambda$; iv) the average number of streams used $\overline{S}$, given by $\overline{S} = \lambda' T_h/\overline{N} = T_h/\overline{T}$. Note that once we know $\overline{N}$ and $\overline{T}$, λ', p_L and $\overline{S}$ are all known.

Requests for a movie arrive according to a Poisson process with rate λ req./hr. Each arrival is independently willing to wait for a random time U minutes; if by then its requested movie is not displayed, it reneges. Let the cumulative distribution for U be $R(u) \triangleq P(U < u)$ and its mean $\overline{U}$. $R(u)$ is referred to as the reneging function. For illustrative purpose, we have considered in this paper the following three reneging functions (note that our study is not limited to these functions):

- Exponential function [4], [5], [7], [11] — Users are always willing to wait for a minimum time $U_{min} \geq 0$, beyond which the (additional) waiting time is exponentially distributed with mean τ, i.e., $R(u) = 0$ if $0 \leq u \leq U_{min}$, and $R(u) = 1 - e^{-(u-U_{min})/\tau}$ otherwise. The average time the users are willing to wait is therefore $\overline{U} = U_{min} + \tau$.
- Linear function [5] — Users are willing to wait for a time U_{min} minutes, after which their waiting time is uniformly distributed between U_{min} and U_{max}. Defining $\Delta \triangleq U_{max} - U_{min}$, we have $\overline{U} = U_{min} + \Delta/2$.
- Step function [15], [16] — Users are willing to wait for a certain amount of time U_0 minutes, beyond which they would not be satisfied and may be considered lost (i.e., reneging function is a step function with a delay limit). We hence have $\overline{U} = U_0$. Owing to its simplicity, it can also be considered as an alternative approximation for many arbitrary reneging functions by setting U_0 to some appropriate value (such as $U_0 = \overline{U}$).

We analyze in the following the auto-gated scheduling, fixed-gating, batch-size based scheme, and the combined scheme, by first considering that λ does not depend on the loss rate.

A. *Auto-gated scheduling*

The probability p_1 that the first user in a batch reneges before the batching window finishes is obviously given by $p_1 = R(W)$. Hence, the average time between consecutive channel allocations $\overline{T}$ is given by, $\overline{T} = \sum_{i=1}^{\infty} p_1^{i-1}(1-p_1)i/\lambda + W$, i.e.,

$$\overline{T} = \frac{1}{(1-R(W))\lambda} + W. \tag{1}$$

Let $\tilde{p}$ be the probability that a request arriving *within* a batching window does not renege by the end of the window. By conditioning on the amount of time left from its arrival until the ending of the batching window, $\tilde{p}$ is given by $\tilde{p} = \int_0^W (1-R(x))dx/W$. Including the first user in the batch, the average batch size is hence given by,

$$\overline{N} = 1 + \lambda W \tilde{p}. \tag{2}$$

Note that $d\overline{N}/dW = \lambda(1-R(W)) \geq 0$; therefore, $\overline{N}$ is non-decreasing in W and attains its maximum when $R(W) = 1$. Hence, to maximize the per-batch profit $(\overline{N} - K)P$, W should be chosen equal to the maximum waiting tolerance of all the users, and the corresponding maximum per-batch profit is $(1 + \lambda\overline{U})P - C$. We see that maximizing per-batch profit makes no sense when the reneging function is exponential, in which case $W = \overline{T} = \infty$, and thus $p_L = 1$. It only makes sense for the step reneging function, for which $W = U_0$ and the result is quite clear. A better measure of profit is the profit rate $\hat{\theta}$. We derive $\hat{\theta}$ with the aforementioned user reneging functions as follows:

- Exponential function — With the above expressions for $\overline{N}$ and $\overline{T}$ (Eqs. (1) and (2)), $\hat{\theta}$ can be obtained as,

$$\hat{\theta} = \begin{cases} \lambda\left(1 - \frac{K}{1+\lambda W}\right), & \text{for } W < U_{min}, \\ \frac{1+\lambda U_{min}+\lambda\tau(1-e^{-W/\tau})-K}{e^{(W-U_{min})/\tau}/\lambda+W}, & \text{otherwise.} \end{cases} \tag{3}$$

We immediately see that the service cannot be profitable if $K > 1 + \lambda(U_{min} + \tau)$. If $1 + \lambda U_{min} \leq K \leq 1 + \lambda U_{min} + \lambda\tau$, the system is profitable when $W \geq W_0 = U_{min} - \tau \ln(1 - (K - \lambda U_{min} - 1)/(\lambda\tau))$. If $K < 1 + \lambda U_{min}$, the system is profitable whenever $W \geq W_0 = (K-1)/\lambda$. The window size maximizing $\hat{\theta}$ is obtained by setting $d\hat{\theta}/dW^* = 0$. Clearly, $W^* \geq U_{min}$.
- Linear function — If $W \leq U_{min}$, p_L is obviously 0 and $\hat{\theta} = \lambda(1 - K/(1+\lambda W))$. If $W > U_{max}$, a batch could not be formed and hence $p_L = 1$ and $\hat{\theta} = 0$. The profit rate can be obtained similarly for $U_{min} < W \leq U_{max}$ with Eqs. (1) and (2).
- Step function — Obviously, if $W > U_0$, we have $p_L = 1$ and $\hat{\theta} = 0$. When $W \leq U_0$, no users would renege ($p_L = 0$) and hence $\hat{\theta} = \lambda\left(1 - \frac{K}{1+\lambda W}\right)$. Obviously, $\hat{\theta}$ increases monotonically with $W \leq U_0$ (with $p_L = 0$) and maximum $\hat{\theta}$ is achieved when $W^* = U_0$. When $W > U_0$, $\hat{\theta} = 0$.

B. *Fixed-gating*

Defining $\tilde{p}$ be the probability that a request would stay till the end of a batching window, we have $\tilde{p} = \int_0^W (1-R(x))dx/W$. The probability that there is no request at the end of a batching window is hence given by $e^{-\lambda\tilde{p}W}$. The average interval between consecutive channel allocations is $\overline{T} = W/(1-e^{-\lambda\tilde{p}W})$, and the average batch size is $\overline{N} = \lambda\tilde{p}W/(1-e^{-\lambda\tilde{p}W})$; from above $\hat{\theta}$ can be obtained.

C. *Batch-size based scheme and Combined scheme*

The analysis of the batch-size based scheme when users do not renege has been studied in [15]. The case for exponential reneging function with $U_{min} = 0$ can be analytically obtained [4]. However, the analysis for other reneging function is more difficult and hence simulation has been used.

In the combined scheme, the parameters M and W under a certain target arrival rate are chosen from the respective batching scheme "optimized" for the rate (the profit rate of the combined scheme is hence no less than either of the two "constituent" schemes). We have used simulation to study the performance of this scheme.

IV. Numerical examples and comparisons

In this section, we present numerical examples and comparisons of the schemes. We first discuss the case of the auto-gated scheduling and show that there is not much difference between the scheme and the fixed-gating. Then we discuss the batch-size based scheme and the combined scheme. (Note that we express λ in req./hr in our plots.)

A. *Window-based schemes*

We first study the auto-gated scheduling with exponential user reneging function with $U_{min} = 0$ and investigate the influence of W on $\hat{\theta}$ and p_L. Then we study the influence of λ, K, τ and U_{min}.

We show in Fig. 2 $\hat{\theta}$ as a function of W ($\tau = 45$ minutes and $\lambda = 100$ req./hr). For a given K, as W increases, $\hat{\theta}$ first increases sharply to reach a maximum, and then decreases slowly. High profit can be achieved with low values of W (especially when K is low). As K increases, more users have to be batched in

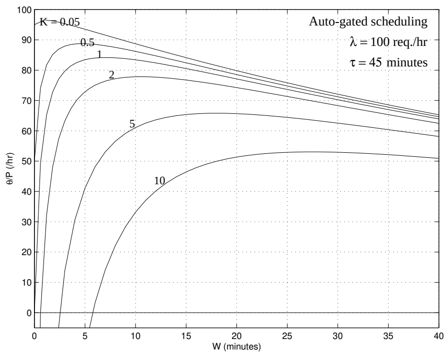

Fig. 2. $\hat{\theta}$ vs. W given K with exponential reneging function.

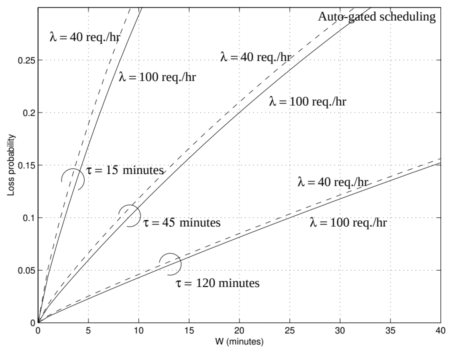

Fig. 3. p_L versus W given λ and τ with exponential reneging function.

order to amortize the channel cost and hence W^* also increases. p_L increases with W as shown in Fig. 3, in which we see that p_L increases rather linearly with W (quite independent on λ). Thus, there is no need to use values of W above W^*. In fact, it may be preferable to choose values for W below W^* in order to keep p_L low. For example, consider $K = 2$; we may choose $W = 4$ minutes (achieving $\hat{\theta} = 70$/hr) instead of the optimum $W^* = 10$ minutes (with $\hat{\theta}^* = 78$/hr), attaining a loss rate of 5% instead of 12%; with $K = 10$, we may choose $W = 12$ minutes (achieving $\hat{\theta} = 40$/hr) instead of $W^* = 27$ minutes (with $\hat{\theta}^* = 51$/hr), attaining a loss rate of about 12% instead of 26%.

We show the influence of λ on W^*, W_0, p_L and $\hat{\theta}$ in Fig. 4. Recall that the region $W < W_0$ is unprofitable while the region $W > W^*$ leads to higher p_L with lower profit. Therefore, in reality it is likely that $W_0 \le W \le W^*$. As λ increases, both W^* and W_0 decrease, first quite sharply and then more gradually. Our result also shows that the p_L's at W^* and W_0 decreases while $\hat{\theta}^*$ increases with λ [23].

We next examine the influence of K on W^* and W_0 in Fig. 5. As K increases, W^*, W_0 and the corresponding p_L's increase with K. We remark here that as K increases, $\hat{\theta}^*$ also decreases quite rapidly [23]. We examine the effect of τ in Fig. 6. Recall that when $K \ge 1 + \lambda U_{min}$, τ has to be larger than a certain value ($= (K-1)/\lambda - U_{min}$) in order to achieve profit. As τ increases, W^* first decreases very sharply and then increases gradually. W_0, on the other hand, decreases with τ, first sharply and then slowly. As τ increases, p_L decreases and $\hat{\theta}^*$ increases. We finally examine the effect of U_{min} in Fig. 7. We see that W^* does not change much for low U_{min}; and as U_{min} increases, it increases keeping $W^* \ge U_{min}$. Note that there is not much change in W_0 as U_{min} increases (in fact, $W_0 = (K-1)/\lambda$ as $U_{min} > (K-1)/\lambda$). As U_{min} increases, p_L's decreases quite fast [23].

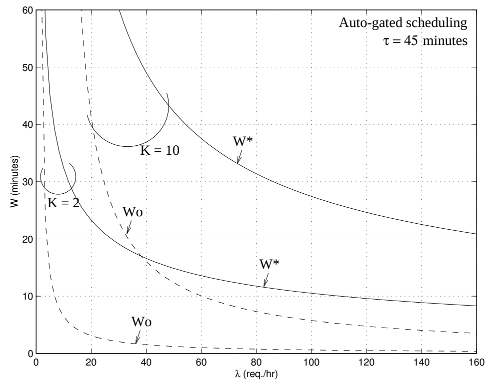

Fig. 4. W_0 and W^* vs. λ given K with exponential reneging function.

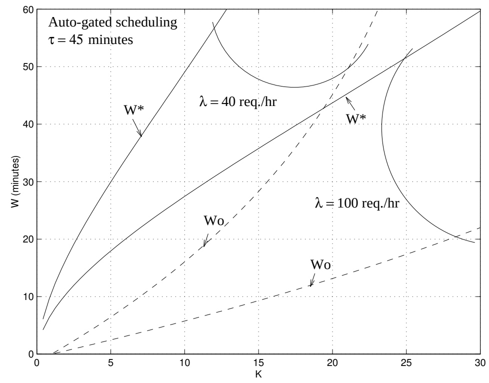

Fig. 5. W_0 and W^* vs. K given λ with exponential reneging function.

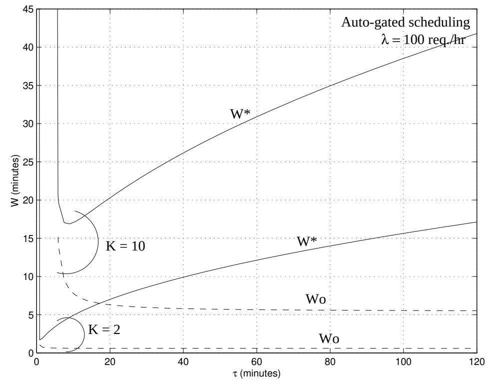

Fig. 6. W_0 and W^* vs. τ given K with exponential reneging function.

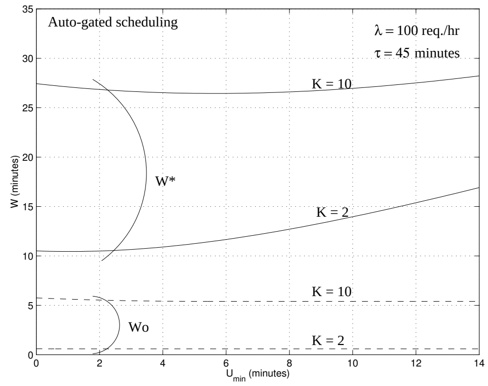

Fig. 7. W_0 and W^* vs. U_{min} given K with exponential reneging function.

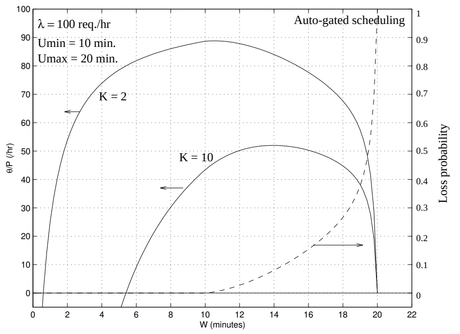

Fig. 8. $\hat{\theta}$ and p_L versus W given K with linear reneging function.

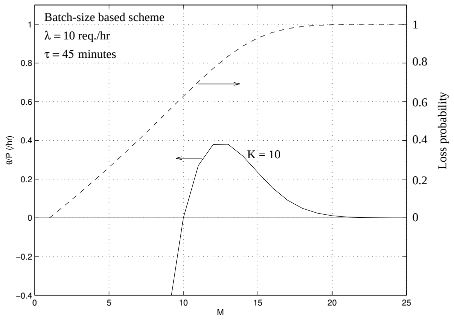

Fig. 10. $\hat{\theta}$ vs. M with $K = 10$ and exponential reneging function.

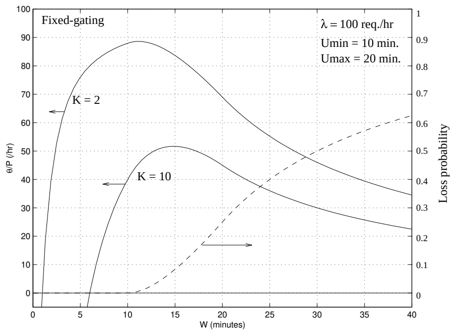

Fig. 9. $\hat{\theta}$ vs. W for the fixed-gating, with linear reneging function.

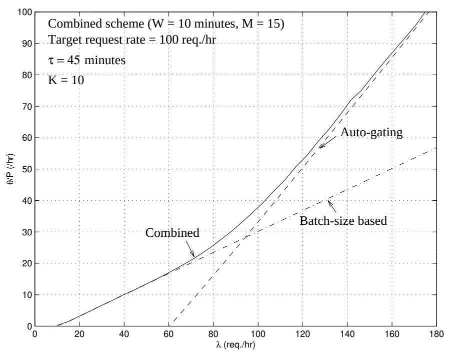

Fig. 11. $\hat{\theta}$ versus λ for the combined scheme, with $\tau = 45$ minutes.

We show in Fig. 8 $\hat{\theta}$ and p_L as a function of W with linear reneging function ($U_{min} = 10$ minutes and $U_{max} = 20$ minutes). For $W \le U_{min}$, $\hat{\theta}$ increases monotonically. For $W \ge U_{min}$, $\hat{\theta}$ first increases (a larger W potentially batches more users) and then decreases rapidly (due to user reneging). We see that the W^*, $\hat{\theta}^*$ and the corresponding p_L are quite similar for the linear and the exponential reneging functions, so long as they have the same mean [23]. (Since the exponential distribution is skewed towards lower delay values than the uniform distribution, W^* and the corresponding $\hat{\theta}^*$ are slightly lower.)

We have studied the fixed-gating, and found that its performance is remarkably similar to that of the auto-gated scheduling with the exponential reneging function [23]. We show in Fig. 9 the case with the linear reneging function. Comparing with the case of the auto-gated scheduling (Fig. 8), we see the remarkable resemblance in $\hat{\theta}^*$ and W^*. However, the fixed-gating is more "forgiving" in performance if W is mis-judged at $W > U_{max}$. The case for the step reneging function show similar conclusions.

B. Batch-size based scheme

For the batch-size based scheme, whenever $M > K$, $\hat{\theta} \ge 0$. There is a batch-size M^* achieving the highest profit rate $\hat{\theta}^*$. When $\hat{\theta}^*$ is high, there is a remarkable resemblance in the $\hat{\theta}^*$ and the corresponding p_L of this scheme and those of the auto-gated scheduling, with the M^* very similar to the $\overline{N}^*$ obtained using W^* in the auto-gated scheduling [23].

There is however a slight difference between the auto-gated scheduling and the batch-size based scheme: While the former does not discriminate between profitable and unprofitable batches, the latter selectively serves a batch which leads to profit (with $M \ge K$). Therefore, when λ is low, the auto-gated scheduling would not be profitable but the batch-size based scheme can be profitable (but with high user loss rate). We illustrate this point in Fig. 10, in which we show $\hat{\theta}$ and p_L as a function of M, given λ and τ. The auto-gated scheduling would not be profitable in this case (Ref. Fig. 4); however the batch-size based scheme maintains the profitability (but with a high loss rate).

C. Combined scheme

We show in Fig. 11 $\hat{\theta}$ versus λ for the combined scheme, with $W = 10$ minutes, $M = 15$ and exponential user reneging function ($U_{min} = 0$, $\tau = 45$ minutes and $K = 10$). Also shown are the performance of the auto-gated scheduling with the same W and the batch-size based scheme with the same M. (Note that the parameters are chosen for each scheme at the target arrival rate $\lambda_0 = 100$ req./hr and $p_L \le\approx 10\%$.) We see that as λ goes higher than the target rate, the auto-gated scheduling achieves higher profit; and when the arrival rate drops, the batch-size based scheme achieves higher profit. The combined scheme traces out the outer "envelop" of the two basic schemes when the actual underlying request rate fluctuates around the target rate. We show in Fig. 12 the corresponding p_L versus λ. We see that the auto-gated scheduling offers a rather uniform p_L as a function of λ, while in the batch-size based scheme p_L can vary significantly with λ. The combined scheme traces out the outer envelop of p_L of the two schemes.

V. Loss-Dependent Arrival Rate

So far, we have considered that λ is independent of p_L. Since the unsatisfied users may never visit the system again, λ can be a function of p_L. In this section, we address how the batching parameters can be specified to achieve maximum profit under this case.

We plot in Fig. 13 p_L versus λ; we show in solid lines the contours achieving a certain profit rate $\hat{\theta}$ (such contours may be obtained from Fig. 4). Note that the iso-$\hat{\theta}$ line increases outwards as shown. The dashed line is an arbitrary loss-dependent arrival rate — we show here that a user would not use a system at all if the loss probability is higher than p_{max}, and $\lambda = \lambda_{max}$

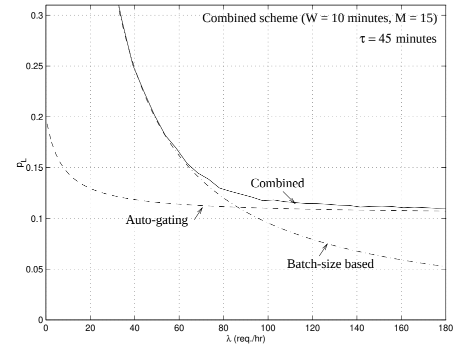

Fig. 12. p_L versus λ for the composite scheme, with $\tau = 45$ minutes.

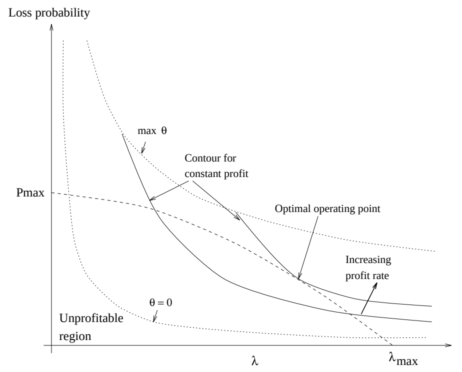

Fig. 13. Maximizing $\hat{\theta}$ with loss-dependent arrival rate.

when $p_L = 0$ (the loss-independent arrival rate is hence a vertical line at λ_{max}). The optimal operating point is that where the iso-$\hat{\theta}$ line touches tangentially the dashed line. We see that if λ is dependent on p_L, the optimal $\hat{\theta}$ is lower, and the loss rate is always less than p_{max}.

VI. Conclusions

A near video-on-demand system achieves high user capacity by batching requests for a particular video and serving them with a single multicast stream. We have considered providing such services when there is a cost associated with using a multicast channel. Batching therefore has to be done so as to amortize such channel usage cost so as to achieve profit.

We have considered how profit can be achieved in providing near VOD services, given certain user reneging behavior. We have studied a number of batching schemes — in the window-based batching schemes, user delay is bounded by a certain maximum value W; and in the batch-size based scheme, revenue can be maintained in each batch by allocating a stream whenever a certain number M of users are collected. We have compared these schemes in terms of their profit rates (which translates into yearly profit) given that users may renege while being batched. When the arrival rate is high, it is advantageous to use the window-based scheme due to its high profit, but when the arrival rate drops, it is advantageous to use the batch-size based scheme to maintain profitability. We have therefore proposed a combined scheme which adaptively operates as above so as to achieve high system profit as the underlying arrival rate fluctuates.

We find that maximizing per-batch profit generally leads to long batching period and high user loss rate. Maximizing profit rate, on the other hand, encourages more frequent smaller returns. However, the loss rate at the maximum profit may still be undesirably high, and hence a shorter batching period should be used in reality. Generally, the higher user delay tolerance is, the longer the batching period is. Our results show that the optimal operating point does not depend very much on the user reneging behavior besides its mean. We find that the maximum profit rate for the window-based scheme and the batch-size based scheme are very similar if both schemes are highly profitable; while the batch-size based scheme is able to maintain profit when the arrival rate is low. If the arrival rate is loss-dependent, the system profit as compared with the loss-independent case is generally lower.

References

[1] T. Little and D. Venkatesh, "Prospects for interactive video-on-demand," *IEEE Multimedia Magazine*, pp. 14–24, Fall 1994.

[2] V. O. Li and W. Liao, "Distributed multimedia systems," *Proceedings of the IEEE*, vol. 85, pp. 1063–1108, July 1997.

[3] F. A. Tobagi, "Distance learning with digital video," *IEEE Multimedia Magazine*, pp. 90–94, Spring 1995.

[4] A. D. Gelman and S. Halfin, "Analysis of resource sharing in information providing services," in *Proceedings of IEEE Globecom'98*, pp. 312–316, 1990.

[5] A. Dan, D. Sitaram, and P. Shahabuddin, "Dynamic batching policies for an on-demand video server," *Multimedia Systems*, vol. 4, pp. 112 – 121, June 1996.

[6] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, "On optimal batching policies for video-on-demand storage servers," in *Proceedings of the 3rd International Conference on Multimedia Computing and Systems*, (Hiroshima, Japan), pp. 253–258, IEEE, 17-23 June 1996.

[7] K. C. Almeroth, A. Dan, D. Sitaram, and W. H. Tetzlaff, "Long term resource allocation in video delivery systems," in *Proceedings of IEEE Infocom'97*, pp. 1333–1340, 1997.

[8] E. L. Abram-Profeta and K. Shin, "Scheduling video programs in near video-on-demand systems," in *Proceedings of ACM International Multimedia Conference*, pp. 359–371, Nov. 9-13 1997.

[9] L. Golubchik, J. C. S. Lui, and R. Muntz, "Reducing I/O demand in video-on-demand storage servers," in *Proceedings of the ACM SIGMETRICS'95*, (Ottawa, Ontario, Canada), pp. 25–36, ACM, 1995.

[10] C. Aggarwal, J. Wolf, and P. S. Yu, "On optimal piggybacking merging policies for video-on-demand systems," *Performance Evaluation Review*, vol. 24, pp. 200–209, May 1996.

[11] S. Sheu, K. A. Hua, and W. Tavanapong, "Chaining: A generalized batching technique for video-on-demand systems," in *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, (Ottawa, Ont., Canada), pp. 110–117, IEEE, 3-6 June 1997.

[12] W. Liao and V. O. Li, "The split and merge protocol for interactive video-on-demand," *IEEE Multimedia*, pp. 51–62, October-December 1997.

[13] M. N. Garofalakis, B. Özden, and A. Silberschatz, "On periodic resource scheduling for continuous media databases," in *IEEE RIDE (Research in data engineering)*, (Orlando, Florida), pp. 111–120, Feb 1998.

[14] A. K. Tsiolis and M. K. Vernon, "Group-guaranteed channel capacity in multimedia storage servers," *Performance Evaluation Review*, vol. 25, no. 1, pp. 285–297, 1997.

[15] S.-H. G. Chan, F. Tobagi, and T.-M. Ko, "Providing on-demand video services using request batching," in *Proceedings of the 1998 IEEE International Conference on Communications (ICC'98)*, (Atlanta, Georgia), pp. 1716–1722, June 7-11 1998.

[16] S.-H. G. Chan, F. Tobagi, and T.-M. Ko, "Bandwidth planning in near video-on-demand," in *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (ISCAS'98)*, (Monterey, CA), pp. 61–64, May 31 - June 3 1998.

[17] P. S. Yu, J. L. Wolf, and H. Shachnai, "Design and analysis of a look-ahead scheduling scheme to support pause-resume for video-on-demand applications," *Multimedia Systems*, vol. 3, no. 4, pp. 137–149, 1995.

[18] K. C. Almeroth and M. H. Ammar, "The use of multicast delivery to provide a scalable and interactive video-on-demand service," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1110–1122, August 1996.

[19] F. A. Tobagi, J. Pang, R. Baird, and M. Gang, "Streaming RAID$^{\text{TM}}$ – a disk array management system for video files," in *Proceedings of First ACM International Conference on Multimedia*, (Anaheim, CA), August 1-6 1993.

[20] R. L. Haskin and F. B. Schmuck, "The tiger shark file system," in *Proceedings of COMPCON'96*, pp. 226–231, IEEE, 1996.

[21] R. Buck, "The Oracle media server for nCube massively parallel systems," in *Proceedings of the 8th International Parallel Processing Symposium*, pp. 670–673, IEEE, 1994.

[22] H. M. Vin and V. Rangan, "Designing a multiuser HDTV storage server," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 153–164, January 1993.

[23] S.-H. G. Chan and F. Tobagi, "Scalable services for video-on-demand." Technical report, CSL-98-773, Computer Systems Laboratory, Stanford University, December 1998.