Tradeoff Between System Profit and User Delay/Loss in Providing Near Video-on-Demand Service

S.-H. Gary Chan, *Member, IEEE,* and Fouad Tobagi, *Fellow, IEEE*

***Abstract*—In a near video-on-demand (near-VOD) system, requests for a movie arriving in a period of time are grouped (or "batched") together and served with a single multicast stream. In this paper, we consider providing near-VOD services when there is a cost associated with using a network multicast channel. We address the tradeoff between system profit, given by the total pay-per-view collected minus the total channel cost, and user delay or user loss (due to reneging). We first analyze and compare the tradeoff of two traditional "basic" schemes, namely, the window-based schemes in which a maximum user delay can be guaranteed, and the batch-size based scheme in which system profit can be guaranteed. By combining these basic schemes, we present a scheme which can adaptively balance system profit and user delay when the underlying request rate fluctuates. We then consider the case in which delayed users may renege and determine how system profit can be maximized by sizing the batching period given user's reneging behavior. We show that maximizing profit can lead to excessively high user loss rate, especially when the channel cost is high and users are not very patient. Therefore, a shorter suboptimal batching period should be used for this case in reality. We finally introduce schemes which are able to offer high profit or low user loss when the underlying arrival rate fluctuates.**

***Index Terms*—Multicasting, near video-on-demand, request batching, system profit, user delay, user loss.**

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, home shopping, distance learning, training, etc. [1]–[3]. In true-VOD, each user is assigned its own dedicated unicast stream (or channel), and hence it enjoys great flexibility in interacting with the server while viewing the video. However, for some applications where a large number of concurrent requests for the same video has to be accommodated, true-VOD becomes very expensive. As an alternative, near-VOD is much more cost-effective, whereby many requests for a given video content arriving over a certain period of time are grouped (i.e., batched) and served with a single multicast stream. Such a scheme is acceptable for applications where user interactivity is not essential, as is the case with movie-on-demand. (A certain degree of user interactivity may be achieved with near-VOD. Readers interested in this aspect are referred to [4]–[6] and the references therein.) Today, request batching is widely used for movie-on-demand services over satellite and cable networks. It simply consists of having the same movie shown at specific prescheduled points in time, with the time between consecutive showings (referred to as the batching window) equal to some fraction of the movie's duration.

Clearly, for a given request arrival process, the larger the batching window is, the smaller is the number of network channels used (and hence the lower is the network cost) and the larger is the batch size (and hence the higher is the revenue per channel). However, this also implies a longer delay experienced by a user (and hence the worse is the quality of the service as compared to an ideal true-VOD system). Since it is quite typical for users to renege after experiencing "long" delays, an excessively long batching period may lead to a high loss in user requests (and thus revenue), and in the long run, loss of customers. Therefore, it is very important in practice to strike a balance between profit and quality of service in terms of user delay and user loss rate. This is the issue we address in this paper.

More specifically, we consider a near-VOD system in which requests to view movies arrive according to a known stochastic process; requests corresponding to the various movies are batched separately over time according to some batching scheme, and each such batch is served by a single multicast stream. We consider in this paper that multicast channels are acquired as needed (i.e., on-demand), and a certain cost is associated with the use of such a channel (the cost is a function of the channel bandwidth and the duration of use). We also consider that each served request is charged a certain pay-per-view fee which is also a function of the movie's characteristics (e.g., its data rate and duration). Our primary interest is to understand, and hence to achieve, the tradeoff between profit (defined as the difference between fees collected and channel-usage cost) and user delay (defined as the time from when a user places a request and the time the movie starts to be displayed).

Concerning user reneging behavior, in the absence of any such information that one could rely on, it is quite appropriate to consider a very simple model in which users are willing to wait for a certain period of time beyond which they get dissatisfied and may be considered as having reneged (i.e., lost). That is, the reneging function is a step function, exhibiting a hard limit on the tolerable delay. A more refined version of this model may be as follows. The user satisfaction is very high if the delay experienced is below a certain value $D_{\min}$, and very

Manuscript received February 10, 2000; revised March 29, 2001. This work was supported in part by the Center for Telecommunications, Stanford University, by the Areas of Excellence (AoE) Scheme on Information Technology funded by the University Grant Council, Hong Kong, and by the Direct Allocation Grant of the Hong Kong Research Grant Council. This paper was presented in part at ICC'98 and ICC'99. This paper was recommended by Associate Editor H. J. Zhang.

S.-H. G. Chan is with the Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: gchan@cs.ust.hk).

F. Tobagi is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA.

Publisher Item Identifier S 1051-8215(01)06529-6.

low if the delay exceeds a certain value $D_{\max}$ (which leads to very high rate of reneging). Delays between $D_{\min}$ and $D_{\max}$ are tolerated (no user reneging), but the user satisfaction is not very high. One could also conceive a number of models for the user reneging behavior, as has been done in the literature: the time that a user is willing to wait before reneging is considered to be distributed according to some cumulative distribution function. Examples of the function are exponential function [7], [8], truncated Gaussian [9], and linear function (corresponding to a uniform distribution) [10]. (We note that the use of any specific model in the literature has been either arbitrary, or driven by the need to keep the underlying analysis tractable.) Where user reneging is assumed, the batching period has an influence on system profit (given by the total pay-per-view collected minus the total channel cost). If it is too short, too much channel cost is incurred; on the other hand, if it is too long, too many users would renege, and hence too little of a revenue is collected. Of interest is to maximize the profit by sizing the batching period, and to achieve the tradeoff between profit and the percentage of users lost (the latter is indicative of user dissatisfaction and would ultimately lead to an attrition on the customer base).

Previous work on near-VOD concentrated on the streaming capacity available at the server. Such capacity was considered to be given, and in a sense already paid for, and therefore of interest was the issue of assigning the available streams to the various requests so as to meet a certain loss rate of requests [7]–[11]. We focus in our study on network multicasting cost, in which case it is important to address the conflicting goals between high system profit and low user delay (or low user loss). Some of the schemes we consider here (the basic schemes) are not new. However, they have been traditionally studied via simulation [10], [8]; we provide their analysis here. More advanced batching techniques, such as piggybacking and client buffering, have been considered in the literature. (Readers interested in that are referred to [12]–[19] and references therein). The schemes presented here can be used in conjunction with theirs to achieve higher bandwidth saving.

This paper is organized as follows. In Section II, we analyze various batching schemes under the assumption that users do not renege (i.e., the batching period of the schemes are within user's delay expectation). We first analyze two well-known basic batching schemes—the window-size based scheme and the batch-size based scheme—and then introduce a new adaptive scheme which combines appropriately the key advantage of the window-size based scheme (namely, guaranteed delay) and the key advantage of the batch-size based scheme (namely, guaranteed per-stream revenue) and therefore adaptively balances system profit and user delay. In Section III, we study the design of near-VOD systems with user reneging by examining the tradeoff between profit and user loss with respect to the batching period. We conclude in Section IV.

II. System Profit and User Delay When Users Do Not Renege

In this section, we consider the case in which users do not renege. We first describe the batching schemes (Sections II-A and II-B), and analyze them in terms of the number of channels required, the number of batched users served by a multicast stream (i.e., the batch-size), the delay experienced by a user, and the tradeoff between system profit and user delay, given a certain multicast channel cost (Section II-C). We finally present some illustrative numerical examples (Section II-D).

The following general considerations apply to the remainder of this paper. We consider a properly designed system in which the probability of running out of multicast channels is sufficiently low and can be ignored. 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. Let the movie duration be T_h minutes, and let P_V (dollars) denote the pay-per-view fee (PPV) charged to the served users.

In near-VOD with multicast channel cost, there is in general a minimum average batch-size K for the system to become profitable. For example, let us consider the cost incurred in using a multicast channel to be the sum of a fixed cost C (dollars) and a cost α (dollars) per user served by that multicast channel. Let N and $\bar{N}$ denote the number of users served in a batch (that is, by a multicast stream) and its average, respectively. Typically N is a random variable the distribution of which depends on the request arrival process for the movie and the particular batching scheme in use. The revenue collected in serving a batch is simply given by NP_V, and the channel cost is given by $C + N\alpha$; thus, serving a batch of size $N = n$ is profitable if $nP_V \geq C + n\alpha$, in which case the profit is given by $n(P_V - \alpha) - C$. Otherwise, a loss is incurred in the amount of $C - n(P_V - \alpha)$. The value of N corresponding to break-even is given by

$$\frac{C}{P_V - \alpha} \triangleq K. \tag{1}$$

Clearly, a batching scheme is profitable if $\bar{N}P_V \geq C + \bar{N}\alpha$, i.e., $\bar{N} \geq K$.

We consider the stochastic process representing the arrival of requests for the movie to have a mean rate of λ requests/minute. Since users do not renege, the rate of revenue is then given by λP_V. The rate of profit, θ (in \$/min), is hence given by $\theta = \lambda P_V - (\lambda/\bar{N})(C + \bar{N}\alpha)$, where the second term on the right hand side is the rate of channel cost. In other words

$$\theta = \lambda P \left(1 - \frac{K}{\bar{N}}\right) \tag{2}$$

where $P \triangleq P_V - \alpha$. From the equation, we see that if users do not renege, the profit is related to $\bar{N}$: the higher $\bar{N}$ is, the higher is the profit. We summarize in Table I the important symbols we use in this paper.

A. Basic Batching Schemes

The basic batching schemes considered in this paper fall into two categories depending on the stopping rule used in batching user requests. In one category, the stopping rule is based on a time-window whereby all users arriving within a well-defined window of time are batched together. In the other category, the stopping rule is based on the number of requests collected. In the following, we describe these basic schemes. We then introduce a batching scheme which combines both stopping rules.

TABLE I
IMPORTANT SYMBOLS USED

T_h	:	Movie length (minutes)
λ	:	External request rate for a movie (req/min)
P_V	:	Pay-per-view (\$)
C	:	The fixed part of channel cost (\$)
α	:	The variable part of channel cost: transmission cost per user (\$)
P	:	$\triangleq P_V - \alpha$ (\$)
K	:	$\triangleq C/P$
$N, \overline{N}, f_N(n)$	:	Batch-size, average batch-size, and batch-size distribution, respectively
$D, \overline{D}, f_D(x)$	:	User delay (minutes), average user delay (minutes), and delay distribution, respectively
W_{max}, W_{min}	:	The maximum and minimum window size for the adaptive scheme, respectively
M	:	Number of users in a batch for the batch-size based scheme
$g_W(w)$	:	Distribution of the batching window for the batch-size based scheme
θ	:	Profit rate (\$/minute)
$\hat{\theta}$	:	Normalized profit rate $\triangleq \theta/P$ (/minute)
λ'	:	Throughput for a movie (req/min)
$\overline{T}$	:	Average interval between successive channel allocation (minutes)
$R(u)$	:	User reneging function given by P(user delay tolerance $\leq u$)

In the time-window based category, we consider three schemes referred to as fixed scheduling, fixed gating, and auto-gated scheduling. The *fixed scheduling* scheme is the simplest: it shows a movie once every exactly W minutes. A user which requests a video between two such showings (the batching window) is served by the next showing following the request. Note that in this scheme, a stream is used even if no request has been made in the batching period preceding it. The *fixed gating* is similar to fixed scheduling, except that a showing is omitted if no user has requested the movie in its corresponding batching window. In the *auto-gated scheduling* (or simply *auto-gating* [20]), a batching window of size W minutes is not preset as in the aforementioned schemes, but started by the first arrival following the ending of the previous batching window.

Note that in all three schemes, the user delay is bounded by W. In both fixed scheduling and fixed gating, the movie showtime may be published in advance, while in the auto-gating, the movie showtime can only be determined, and thus advertised, at the start of the corresponding batching window. In fixed scheduling, the number of concurrent streams required is deterministic and given by $\lceil T_h/W \rceil$, while in fixed and auto-gating, the number of concurrent streams required is random and depends on the arrival process. Finally, in all the three schemes, the batch size, and thereof the profit, is random and depends on the requests' arrival process. When the arrival rate is high, the batch size is large and high profit may be achieved. Conversely, when the arrival rate drops, the batch size also drops and profitability may no longer be guaranteed.

The *batch-size based scheme* we consider here is quite simple. The batching period ends when the batch-size has reached a certain value M. Clearly by setting $M \geq K$, profitability is guaranteed. However, the user delay is not bounded and depends on the arrival process, and the movie showtime remains unknown and cannot be advertised to the users prior to the showtime.

B. Adaptive Scheme

We now introduce a new adaptive scheme which combines the key advantage of the window-size based schemes (namely guaranteed delay) and the advantage of the batch-size based scheme (namely guaranteed per stream revenue) by ensuring that when the arrival rate is sufficiently high (and hence, profit can be easily achieved), the system guarantees fairly low delay to the users. But when arrival rate is not so high, the system guarantees certain profit as long as user's delay does not exceed a certain bound. The scheme therefore balances adaptively service quality (in terms of the delay user experienced) and system profit.

We consider that user satisfaction is high if the delay experienced is below a certain value $D_{\min}$. We also consider that there is a delay $D_{\max} > D_{\min}$ beyond which user satisfaction is very low and, for all practical purposes, users should not be delayed beyond that. Delays between $D_{\min}$ and $D_{\max}$ are tolerated (i.e., no reneging is likely to take place), but the user satisfaction is not high. As an example, $D_{\min}$ may be in the range 3–7 minutes while $D_{\max}$ may be in the range 15–40 minutes.

According to this user satisfaction model, the adaptive scheme has three parameters: $M \geq K, W_{\min} = D_{\min}$, and $W_{\max} = D_{\max}$, and operates as follows. A batching window is started upon the arrival of the first request after a movie showing. If M users arrive within $W_{\min}$, the system keeps batching until $W_{\min}$ is reached, thereby increasing the profitability beyond the minimum M; if $W_{\min}$ is reached before M users are collected, the batching window is extended until either M or $W_{\max}$ is reached, whichever occurs first. Thus, when the arrival rate drops, the system tries to maintain profitability by using the batch-size based scheme with $M \geq K$; but since users should not be delayed beyond $D_{\max}$, a maximum batching window of $W_{\max}$ is imposed, i.e., even if there are fewer than M users arriving within the window of size $W_{\max}$, the movie is shown anyway. Note that users may not know the exact video show time; however, the show time is guaranteed to be in the range $(W_{\min}, W_{\max})$ following the first arrival in the batch.

C. Scheme Analysis

In analyzing the batching schemes described above, we consider, for the sake of simplicity, that the request arrival process for the movie is Poisson with rate λ requestsmin. Let $\bar{S}$ denote the average number of concurrent streams used for the movie. Clearly, for true VOD, we have $\bar{S} = \lambda T_h$. Recall that N is the

the batch size and $\bar{N}$ is its mean; we let $f_N(n)$ denote its distribution. Since there is no user reneging, $\bar{N}$ is related to $\bar{S}$ by $\lambda T_h = \bar{S}\bar{N}$. Let D denote the delay of a user, with $f_D(t)$ its distribution and $\bar{D}$ its mean. Explicit expressions of these parameters for the various schemes are quite easy to derive as follows.

1) *Fixed Scheduling:* The distribution of the batch size is Poisson with mean $\bar{N} = \lambda W$. Thus, $\bar{S} = T_h/W$. D is uniformly distributed between 0 and W, and hence $\bar{D} = W/2$.

2) *Fixed Gating:* The probability that there are i $(i \geq 1)$ requests in a batch is given by $(\lambda W)^i e^{-\lambda W}/(i!(1-e^{-\lambda W}))$ (a truncated Poisson distribution); hence, $\bar{N} = \lambda W/(1 - e^{-\lambda W})$ and $\bar{S} = T_h(1 - e^{-\lambda W})/W$. Given that there is an arrival within a batching window W, its arrival time is uniformly distributed within the window, i.e., $f_D(t) = 1/W$, for $0 \leq t \leq W$, and hence $\bar{D} = W/2$.

3) *Auto-Gating:* Clearly, the distribution of the batch size is given by $P(N = i) = (\lambda W)^{i-1}e^{-\lambda W}/(i-1)!$, for $i \geq 1$ (a Poisson distribution offset by 1). Therefore

$$\bar{N} = 1 + \lambda W \tag{3}$$

and $\bar{S} = \lambda T_h/(1 + \lambda W)$. (It can be easily shown that the difference in $\bar{N}$ among all the window-based schemes is at most one.) In terms of the delay distribution, since the first user in each batch experiences delay W while the remaining ones in the batch have delay uniformly distributed between 0 and W, we have

$$f_D(t) = \frac{1}{1+\lambda W}\delta(t - W) + \frac{\lambda}{1+\lambda W}, \quad \text{for } 0 \leq t \leq W \tag{4}$$

where $\delta(t)$ is the impulse function with $\delta(t) = 0$ for $t \neq 0$ and $\int_{-\infty}^{\infty} \delta(t)\, dt = 1$. Hence, $\bar{D} = (\lambda W + 2)/(2(1 + \lambda W))W$. Clearly, as λ increases, the impulse at W decreases, and the delay distribution approaches a uniform distribution with mean $W/2$.

4) *Batch-Size Based Scheme:* N is deterministic in this case and equal to M, and $\bar{S} = \lambda T_h/M$. Let W denote the batching period, which is a random variable equal to the sum of $(M - 1)$ exponential variables; therefore, its distribution $g_W(w)$ is given by

$$g_W(w) = \frac{\lambda(\lambda w)^{M-2}}{(M-2)!}\, e^{-\lambda w} \tag{5}$$

and its mean is given by $(M - 1)/\lambda$. The user delay distribution is obtained by conditioning on W as follows. Given $W = w$, the first user in the batch has delay w, the last user has delay equal to 0, while the remaining $M - 2$ users have delay uniformly distributed between 0 and w, i.e., $f_D(x \mid w) = (1/M)\delta(x-w)+(M-2)/(M)(1/w)+(1/M)\delta(x)$, for $0 \leq x \leq w$. Removing the condition on w by using (5), the user delay distribution is given by $(x \geq 0)$

$$f_D(x) = \frac{1}{M}g_W(x) + \frac{M-2}{M}\int_x^{\infty} \frac{g_W(w)}{w}\, dw + \frac{\delta(x)}{M} \tag{6}$$

and $\bar{D} = (M - 1)/(2\lambda)$, which is half of the average batching period (as expected).

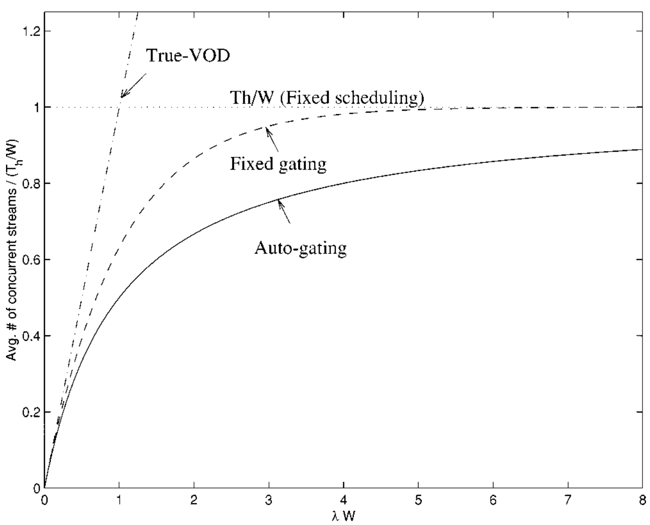

Fig. 1. Comparison of various window-size based schemes in terms of $\bar{S}$ normalized to T_h/W versus λW.

The detailed analysis for the adaptive scheme is more involved, and is shown in Appendix A.

D. Numerical Results and Comparisons

In this section, we present some illustrative numerical results of the schemes. We consider the window-size based schemes first, followed by the batch-size based scheme and adaptive scheme.

We first compare true-VOD, fixed scheduling, fixed gating, and auto-gating in terms of their stream requirement $\bar{S}$. We plot in Fig. 1 $\bar{S}$ normalized to the maximum number of streams used (i.e., T_h/W) as a function of λW for these schemes. Clearly, for a given W, as λ increases, $\bar{S}$ for the true-VOD case increases without bound, while all the window-size based schemes approach a limiting value given by T_h/W. Fixed scheduling always consumes a fixed T_h/W number of streams no matter what the arrival rate is; and for $\lambda \leq 1/W$, it has even higher $\bar{S}$ than true-VOD. Auto-gating achieves the lowest $\bar{S}$ among all the schemes. It is not hard to show that fixed-gating consumes up to 30% more streams than auto-gating (attained at $\lambda W = 1.7934$), while the maximum difference in $\bar{S}$ between the two is 0.2036 T_h/W (attained at $\lambda W = 2.512\,76$). In the following, we limit ourselves to auto-gating and use $T_h = 90$ minutes.

We first consider the maximum delay incurred by a user when a certain level of system profit as given by $\bar{N}_0$ is to be guaranteed ($\bar{N}_0 \geq K$ in order to achieve profit). Recall that the maximum user delay is simply given by the window size W, which is selected according to $1 + \lambda W = \bar{N}_0$ (clearly, this assumes that λ is known *a priori*). We show in Fig. 2 the maximum user delay versus λ to achieve various values of $\bar{N}_0$. As λ increases, the maximum user delay decreases. As λ decreases, the user delay has to increase very rapidly in order to maintain the same level of profit.

Regarding the batch-size based scheme, since the batching period is extended until exactly M users have been collected, the delay in the batch-size based scheme is not bounded as is the case with the window-size based schemes. A typical distribution of user delay is shown in Fig. 3 ($\lambda = 25$ requests/h and $M = 5$).

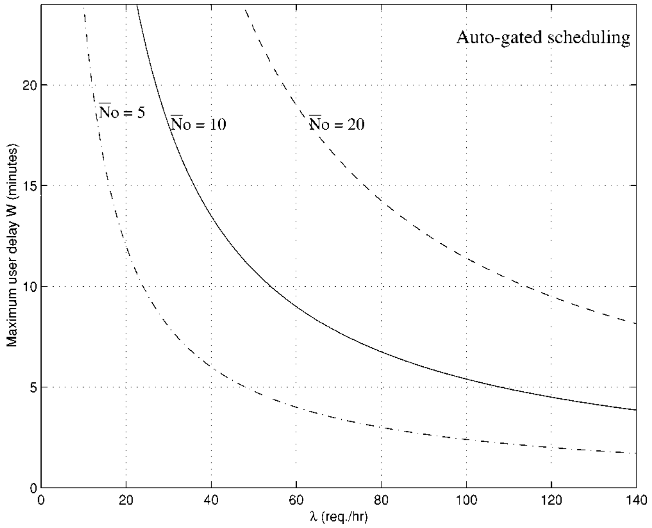

Fig. 2. Maximum user delay given by W versus λ to meet a certain profit level indicated by $\bar{N}_0$.

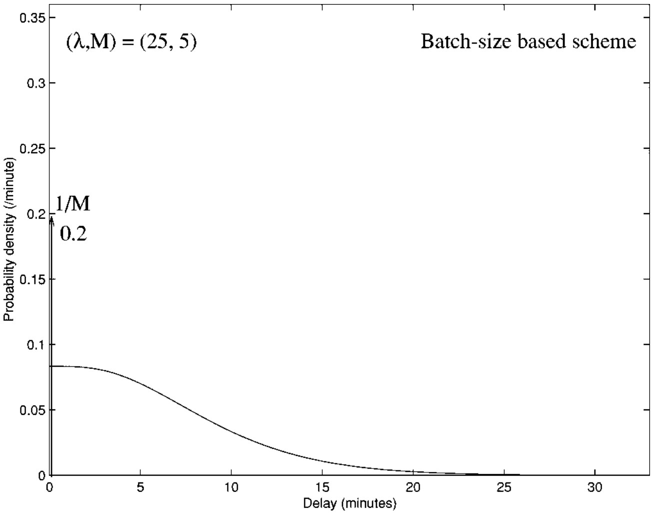

Fig. 3. Delay distribution for the batch-size based scheme (λ = 25 requests/h, $M = 5$).

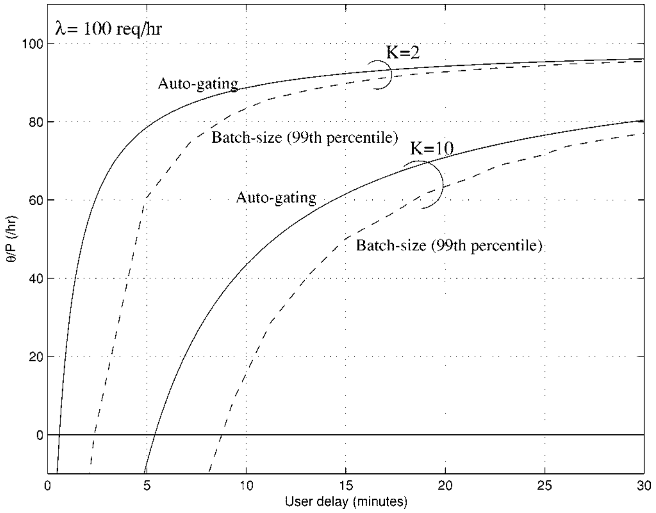

Fig. 4. Profit given by θ/P versus maximum user delay for the basic schemes (99th percentile for the batch-size based scheme), for $K = 2$ and 10.

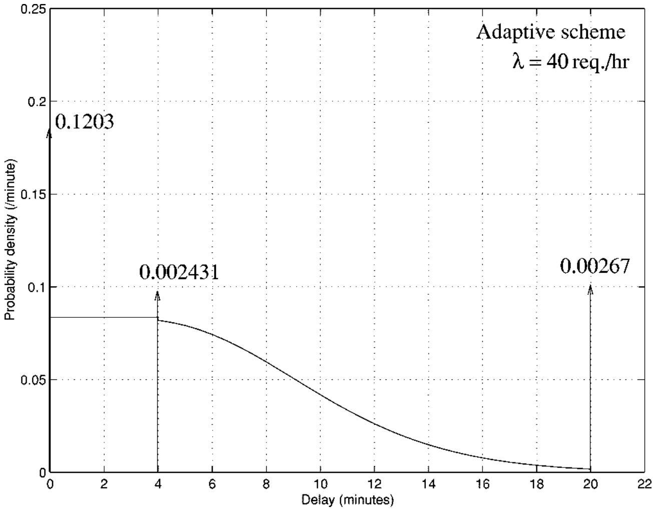

Fig. 5. User delay distribution for the adaptive scheme with $\lambda = 40$ requests/h, and $(W_{\min}, M, W_{\max}) = (4, 8, 20)$.

There is an impulse at the origin with magnitude $1/M$, since the last user in every batch enjoys zero delay. For a given M, when λ decreases, the impulse at the origin does not change but the distribution spreads out, indicating an increase in the user delay. When M increases (for a given λ), the impulse in the origin decreases while the tail spreads.

We now examine the tradeoff between profit and user delay for both auto-gating and the batch-size based scheme. Regarding user delay, we use maximum delay for auto-gating (i.e., W) and the 99th percentile delay for the batch-size based scheme. We show in Fig. 4 θ normalized to P versus user delay for $K = 2$ and 10 ($\lambda = 100$ requests/h). Clearly, θ increases when users experience higher delay. It first rises rather sharply from negative values (due to high channel cost) to positive values (due to the channel cost amortized by PPV collected), and approaches asymptotically to λP (note that the system revenue is given by λP_V). If K is low, the profit can be very close to the maximum with low user delay; on the other hand, if K is high (e.g., $K = 10$), high profit may not be achieved without incurring long user delay. Given a certain maximum user delay not to be exceeded, the profit for the batch-size based scheme is lower than that for auto-gating, and the difference is significant when the delay requirement is low. Combining with its deterministic delay, auto-gating appears to be more attractive than batch-size based scheme.

We finally consider the adaptive scheme. In Fig. 5, we show a typical distribution of user delay ($\lambda = 40$ requests/h). We see that indeed no user experiences a delay higher than $W_{\max}$; there are some with delay $W_{\min}$, some with delay $W_{\max}$, and some with zero delay (those ending the batching period by making up a batch of M users). When the arrival rate is low, the distribution is similar to that of the auto-gating with window size $W_{\max}$; on the other hand, when λ is high, the distribution is similar to that of the auto-gating with window size $W_{\min}$.

In Fig. 6, we show in solid line the profit θ/P versus λ for the adaptive scheme with $K = 5, W_{\min} = 4$ minutes, $M = 8$, and $W_{\max} = 20$ minutes. We clearly see that when the arrival rate increases, the scheme first uses a larger window size (by

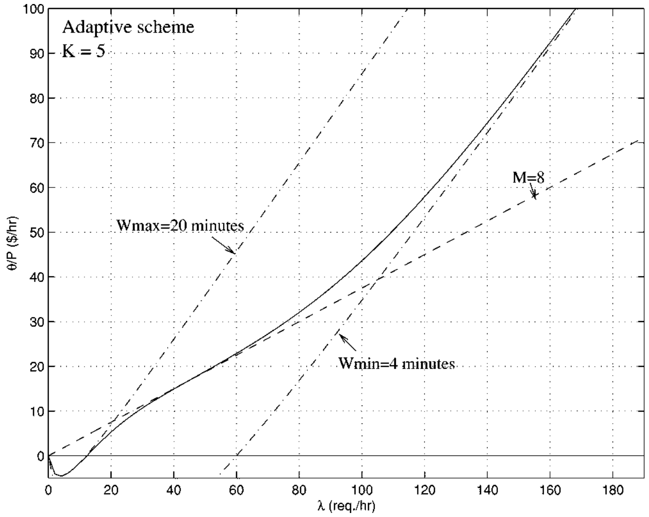

Fig. 6. Profit θ/P versus λ for the adaptive scheme ($W_{\min} = 4$ minutes, $W_{\max} = 20$ minutes, $M = 8$, and $K = 5$). Also shown in dashed lines are θ/P for auto-gating with $W = 4$ minutes and $W = 20$ minutes, and for the batch-size based scheme with $M = 8$.

following the auto-gating curve with $W_{\max} = 20$ minutes) so that the system can recover from a loss. As λ further increases, the adaptive scheme offers a better service by switching to a batch-size scheme (with $M = 8$) and then to the auto-gating with a smaller window size (of $W = 4$ minutes).

We next examine the delay of the adaptive scheme and show in Fig. 7 the corresponding 99th percentile user delay versus λ. When λ is low, user delay is bounded by $W_{\max}$. When λ is high, most of the users enjoy low delay no longer than $W_{\min}$. And for intermediate values of λ, most of the user delay is between $W_{\min}$ and $W_{\max}$ following the curve corresponding to the batch-size based scheme.

III. Achieving the Tradeoff Between System Profit and User Loss With User Reneging

In this section, we consider the tradeoff between profit and user loss when users may renege from the system. We first consider the auto-gating and then two schemes based on the combination of auto-gating and batch-size based scheme, given the network cost model mentioned in Section II. We are interested in the following closely related measures:

1) average per-batch profit, obviously given by $(\bar{N} - K)P$;
2) profit rate θ (the profit per unit time ($/minute)), given by

$$\theta = \frac{(\bar{N} - K)P}{\bar{T}} \tag{7}$$

where $\bar{T}$ is the average period between two consecutive movie showtimes, and the normalized profit rate with respect to $P, \hat{\theta}$ (i.e., $\hat{\theta} \triangleq \theta/P$);
3) system throughput $\lambda^{'}$, representing the number of requests served per minute, given by $\lambda^{'} = \bar{N}/\bar{T}$.

Clearly, the revenue is $\lambda^{'} P_V$ ($/min) and the loss rate p_L is given by $p_L = 1 - \lambda^{'}/\lambda$. Note that once $\bar{N}$ and $\bar{T}$ are known, $\theta, \lambda^{'}$ and p_L are all known.

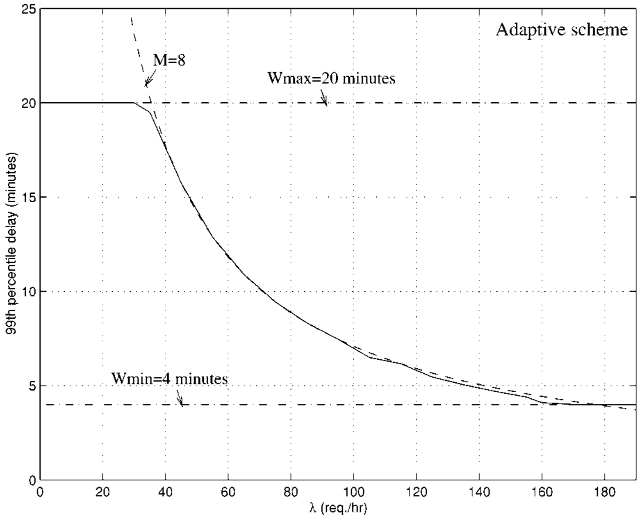

Fig. 7. 99th percentile user delay versus λ for the adaptive scheme ($W_{\min} = 4$ minutes, $W_{\max} = 20$ minutes, and $M = 8$). Also shown in dashed lines are the corresponding delay for auto-gating and batch-size based scheme.

A. Users' Reneging Behavior

As before, we consider that requests for a movie arrive according to a Poisson process with rate λ requestsmin. The waiting tolerance of the users is independent of each other, and each is willing to wait for a period of time $U \geq 0$ minutes; if its requested movie is not displayed by then, it reneges. (Note that even if the start time of a movie is known, a user may lose its interest in a movie and cancel its request if it is delayed too long; in this case, the user is defined lost or "reneged.") U is a random variable with its cumulative distribution denoted by a the user reneging function $R(u) = P(U < u)$ and a mean denoted by $\bar{U}$ minutes. We consider the following exponential reneging function. Users are always willing to wait for a minimum time $U_{\min} \geq 0$; the additional waiting time beyond $U_{\min}$ minutes is exponentially distributed with mean τ minutes, i.e.,

$$R(u) = \begin{cases} 0, & \text{if } 0 \leq u \leq U_{\min} \\ 1 - e^{-(u - U_{\min})/\tau}, & \text{otherwise.} \end{cases} \tag{8}$$

Obviously, the larger τ is, the more delay users can tolerate. Take $U_{\min} = 0$ as an example. With $\tau = 15$ minutes, users are not very patient (almost 30% of the users cannot wait beyond 5 minutes and more than 60% of the users cannot wait beyond 15 minutes), while with $\tau = 120$ minutes, users are very patient (almost 80% of the users can tolerate beyond 30 minutes). Note that since users are more willing to wait in the window-based scheme than the batch-size based scheme, the reneging function for the window-based schemes would have a longer tail. We have also considered linear and step reneging functions, but the results are very similar, and hence would not be presented here [21].

B. Basic Schemes: Profit Analysis

We first consider the profit issues for window-size based schemes. For auto-gating, the first user of a batch is forced to wait W minutes (the window size) before it is served. Note that if the first user in a batch reneges, the batching window is advanced to the next request. (Clearly, W should be less than the maximum tolerable waiting time of the users; otherwise, a batch could never be possibly formed.)

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

$$\bar{T} = \frac{1}{(1-R(W))\lambda} + W. \tag{9}$$

Let $\tilde{p}$ be the probability that a request arriving *within* a batch remains till the end of the batching window. Given that a request arrives within the batching window W, its arrival time is uniformly distributed in the window. By conditioning on the amount of time left from its arrival until the batching window ends, $\tilde{p}$ is given by

$$\tilde{p} = \int_0^W (1-R(x))\frac{dx}{W}. \tag{10}$$

Including the first user in the batch, the average batch size is hence given by

$$\bar{N} = 1 + \lambda W \tilde{p}. \tag{11}$$

Note that $d\bar{N}/dW = \lambda(1-R(W)) \geq 0$. Therefore, $\bar{N}$ is a nondecreasing function in W, and attains its maximum when $R(W) = 1$. Hence, to maximize the per-batch profit $(\bar{N} - K)P, W$ should be chosen equal to the maximum tolerable delay of the users, and the corresponding maximum per-batch profit is then given by $(1+\lambda\bar{U})P - C$. Maximizing per-batch profit, therefore, makes no sense when the reneging function is exponential, since then $W = \infty$, and thus $p_L = 1$. A more interesting measure is to maximize the profit *rate*, which encourages small frequent return by examining the system profit over the infinite time horizon.

Applying the above expressions of $\bar{N}$ and $\bar{T}$ to the exponential reneging function, we obtain the following expression for the normalized profit rate $\hat{\theta}$, we have (12), shown at the bottom of the page. Using the above, it is not difficult to show that the system cannot be profitable (i.e., $\hat{\theta} < 0$) if $K > 1+\lambda(U_{\min}+\tau)$. For $1+\lambda U_{\min} \leq K \leq 1+\lambda U_{\min}+\lambda\tau$, the system is profitable when

$$W \geq W_0 = U_{\min} - \tau \ln\left(1 - \frac{K-\lambda U_{\min}-1}{\lambda\tau}\right). \tag{13}$$

For $K < 1+\lambda U_{\min}$, the system is profitable whenever $W \geq W_0 = (K-1)/\lambda$ [21]. The window size which maximizes $\hat{\theta}$ is obtained by setting $d\hat{\theta}/dW^* = 0$. Clearly, $W^* \geq U_{\min}$. In Appendix B, we derive the user delay distribution for such a reneging function.

We next consider fixed gating, whose analysis is similar to that of auto-gating. Define $\tilde{p}$ to be the probability that a request would stay until the end of a batching window. By conditioning on the arrival time of the request, 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 $e^{-\lambda\tilde{p}W}$. Therefore, the average interval between consecutive channel allocations is $\bar{T} = W/(1-e^{-\lambda\tilde{p}W})$, and the average batch size is $\bar{N} = \lambda\tilde{p}W/(1-e^{-\lambda\tilde{p}W})$. From these, $\hat{\theta}$ can be obtained.

We next consider the batch-size based scheme. In this scheme, the system continues to batch requests until a certain number M of requests are collected (The system is aware of users reneging and does not count users that have reneged). If M is too low, then the profit is low (due to too small a number of users collected and too high a number of channels allocated over time); however, if M is too high, the profit is also low (due to user reneging). Therefore, we expect that there is an optimal value of M which maximizes the profit rate. The mathematical analysis of the scheme with arbitrary user reneging functions is difficult, and we have used simulation to study it.

C. Combining Auto-Gating and Batch-Size Based Scheme

We now consider combining auto-gating and the batch-size based scheme so as to achieve either high profit or low loss when the arrival rate fluctuates around a given target value. The schemes are similar to the adaptive scheme mentioned in Section II-B, but with $W_{\max} = \infty$ and some modifications due to user reneging.

In the *combined-for-profit* scheme, requests are batched until the number of users collected (excluding those reneged) is no fewer 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 showtime) is no less than W minutes (so as to safeguard against too short a batching period).[1] We see that the scheme operates according to auto-gating when the arrival rate is high, and according to the batch-size based scheme when the rate drops. The profit rate of the combined scheme is hence no less than either of the two "constituent" schemes. Since there is a tradeoff between profit and user loss, the cost in achieving high profit this way is higher user loss.

On the other hand, in the *combined-for-loss* scheme, the loss rate is kept low at the sacrifice of some profit: the scheme keeps batching users until the first user in the batch has waited for W minutes *or* the number of users collected is M, whichever is earlier. Obviously, such a scheme operates as auto-gating when the arrival rate is low, and the batch-size based scheme when the arrival rate increases.

The appropriate batching parameters for the above schemes, namely W and M, are chosen from its respective constituent

$$\hat{\theta} = \begin{cases} \lambda\left(1 - \frac{K}{1+\lambda W}\right), & \text{for } W < U_{\min} \\ \dfrac{1+\lambda U_{\min} + \lambda\tau(1-e^{-(W-U_{\min})/\tau}) - K}{e^{(W-U_{\min})/\tau}/\lambda + W}, & \text{otherwise.} \end{cases} \tag{12}$$

[1] Note that $W_{\min}$ in the previous adaptive scheme is called W in this scheme.

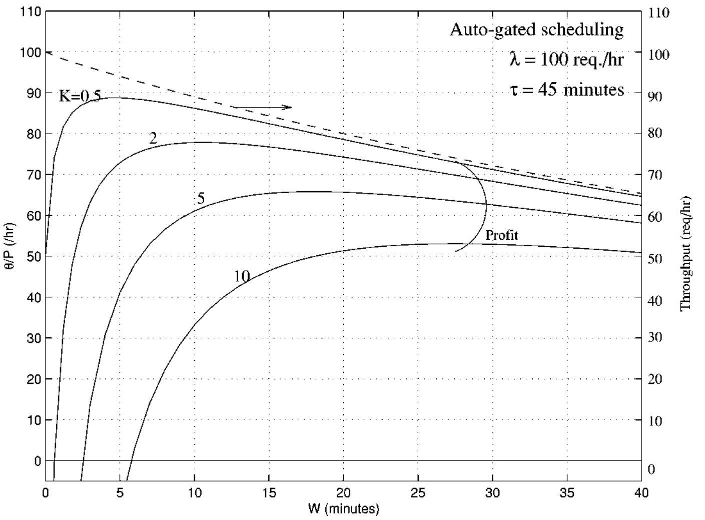

Fig. 8. $\hat{\theta}$ and λ' versus W given K with exponential reneging function ($U_{\min} = 0$ and $\tau = 45$ minutes).

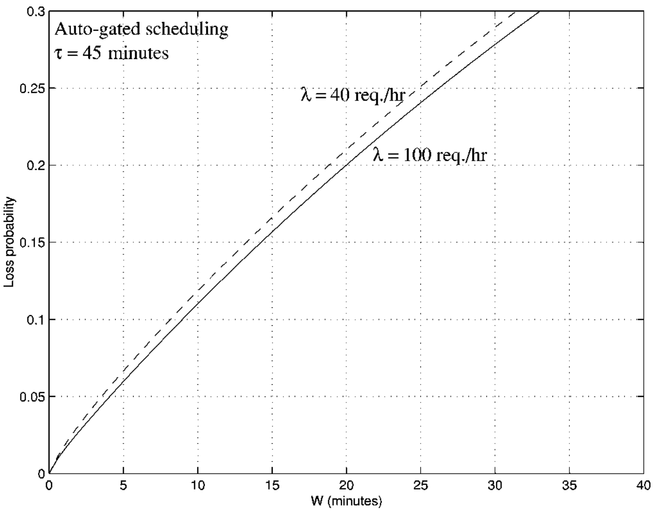

Fig. 9. p_L versus W given λ for auto-gating.

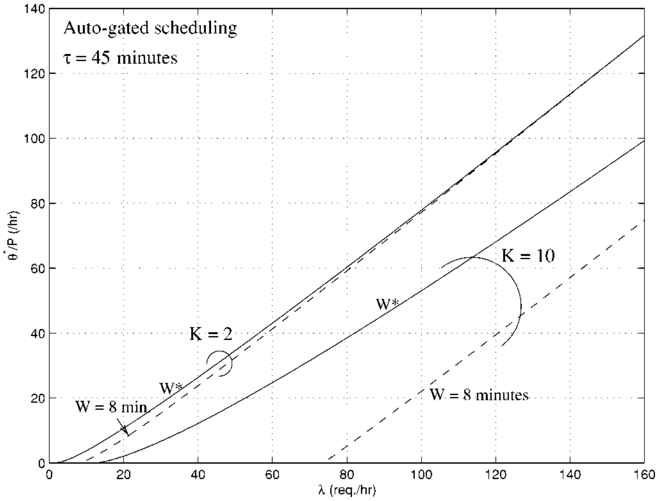

Fig. 10. $\hat{\theta}^*$ versus λ given K ($\tau = 45$ minutes). Also shown in dashed lines are $\hat{\theta}$ corresponding to $W = 8$ minutes (achieving p_L ≈9%).

batching schemes (i.e., W in the window-based scheme and M in the batch-size based scheme) "optimized" independently for the target arrival rate for the movie, i.e., to achieve the maximum profit subject to a user loss requirement when the other is absent. We have used simulation to study the performance of this scheme.

D. Numerical Results

In this section, we present numerical results for the batching schemes given the exponential user reneging function. We first discuss the window-size based schemes, focusing on auto-gating (The performance of fixed-gating is similar to that of the auto-gating and will not be presented here [21]). Then we discuss the batch-size based scheme and the combined schemes. We are primarily concerned with achieving maximum profit subject to a certain user loss requirement (say around 10%).

For auto-gating, we first study the influence of W on $\hat{\theta}$ and p_L, with $U_{\min} = 0$, and $\tau = 45$ minutes. We show in Fig. 8 $\hat{\theta}$ (in solid lines) and system throughput $\lambda' = \lambda(1 - p_L)$ (in dashed line) as a function of W ($\lambda = 100$ requests/h and $\tau = 45$ minutes). Clearly, λ', and hence the revenue rate, decrease with W due to user reneging. The profit $\hat{\theta}$, however, first increases rather sharply to reach a maximum, and then decreases rather slowly as W increases (asymptotically to the value given by $\lambda' P$, as expected). Maximum profit can be achieved with low values of W, especially when K is low. As K increases, more users have to be batched in order to amortize the channel cost and hence W^* increases. As W increases, p_L also increases as shown in Fig. 9, indicating that there is no point to use values of W beyond W^*. It is worth noting that p_L increases rather linearly with W, and does not depend much on λ. For example, to maintain a loss rate at around 10% for λ ranging from 40 to 100 requests/h, we can use $W = 8$ minutes. Figs. 8 and 9 show that, for the range of W between 0 and W^*, there is a tradeoff between profit and user loss, and it may be necessary to choose values of W below W^* in order to keep p_L low, especially when K is large. For example, consider $K = 2$; we may choose $W = 4$ minutes (achieving $\hat{\theta} = 70$/h) instead of the optimum $W^* = 10$ minutes (with $\hat{\theta}^* = 78$/h) to decrease the loss rate from 12% to 5%; with $K = 10$, we may choose $W = 12$ minutes (achieving $\hat{\theta} = 40$/h) instead of $W^* = 27$ minutes (with $\hat{\theta}^* = 51$/h) to decrease the loss rate from 26% to about 12%.

We show in Fig. 10 the maximum profit (i.e., with $W = W^*$) $\hat{\theta}^*$ in solid lines as a function of λ given K. The maximum attainable profit increases somewhat linearly with λ. Furthermore, for a given K, there is a minimum value of λ for the system to become profitable (the minimum λ is given by setting the maximum per-batch profit to zero, i.e., $1 + \lambda\bar{U} = K$). Also shown in dashed lines is the profit with $W = 8$ minutes, corresponding to a loss rate of about 9%. We see that for low K (such as $K = 2$), even though W^* varies with λ, keeping W constant can achieve close to optimal profit. On the other hand, when K is high (say $K = 10$), meeting a loss rate requirement would mean a more substantial decrease in profit. We see that if the desire is to achieve a low loss rate (say around 10%), then the window size is likely to be sub-optimal, unless

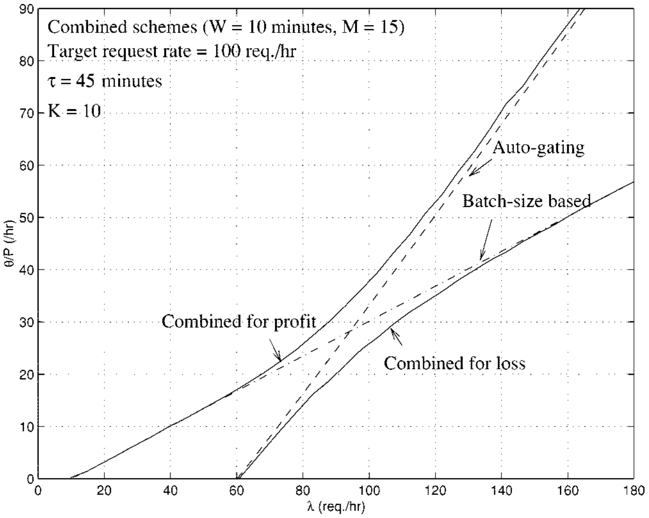

Fig. 11. $\hat{\theta}$ versus λ for the combined schemes ($\tau = 45$ minutes).

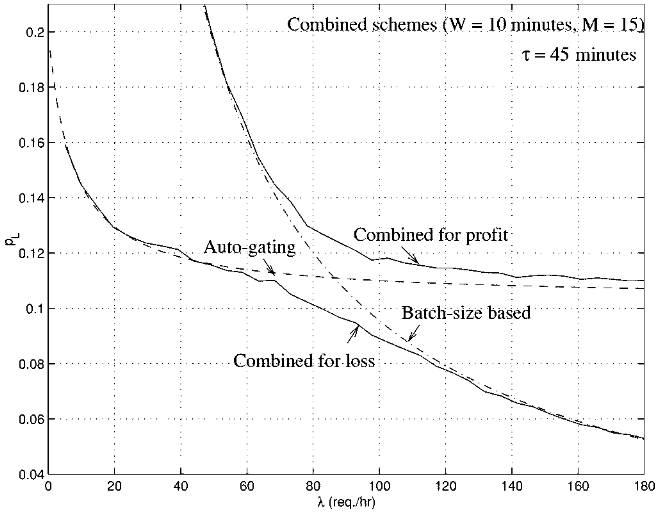

Fig. 12. p_L versus λ for the combined schemes ($\tau = 45$ minutes).

the request rate is high. The figure suggests that movies with low arrival rates may not be profitable if the loss requirement has to be kept low. In order to achieve overall system profit, either the loss requirement of these unpopular movies has to be relaxed, or their PPVs have to be higher, otherwise their loss may have to be offset by the profit from the more popular movies. We also find that (not presented here) if users are willing to wait longer (i.e., as $\bar{U}$ or τ increases), the overall profit increases and the loss rate decreases (with an increase in the optimal window size). This suggests that for those unpopular movies, some means to extend the viewers waiting tolerance may be useful (by offering, for example, deterministic delay or some delay-based PPV).

We next consider the batch-size based scheme. In this scheme, there is an optimal batch size M^* to achieve maximum profit rate $\hat{\theta}^*$. When the profit is high, there is a remarkable resemblance between this scheme and auto-gating in terms of $\hat{\theta}^*$ and the corresponding p_L, with the optimal batch-size M^* corresponds closely to the average batch-size obtained in the auto-gating using W^* [22]. There is, however, a slight difference between the auto-gating 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 (by simply setting $M \geq K$). Therefore, when λ is low, the auto-gating may not achieve profit while the batch-size based scheme can still be profitable (at the expense of higher user loss rate) [21].

We finally consider the combined schemes, and show in Fig. 11 $\hat{\theta}$ versus λ with $W = 10$ minutes, $M = 15$ and exponential user reneging function ($K = 10$, $\tau = 45$ minutes and $U_{\min} = 0$). The parameters W and M are independently chosen given the target arrival rate at $\lambda_0 = 100$ requests/h to achieve $p_L \leq \approx$10% for the auto-gating (Figs. 8 and 9) and batch-size based scheme, respectively. Also shown in broken lines are the respective performance of auto-gating with the same W and the batch-size based scheme with the same M. Clearly, as λ goes higher than the target rate, auto-gating achieves higher profit, and when the arrival rate drops, the batch-size based scheme achieves higher profit. The combined-for-profit scheme traces out the outer "envelope" of the two basic schemes, while the combined-for-loss scheme traces out the lower envelope. We show in Fig. 12 the corresponding p_L versus λ. We see that the combined-for-profit scheme achieves its high profit at the expense of user loss, while its counterpart the combined-for-loss scheme achieves a lower user loss at the expense of profit. The figure also shows that the auto-gating offers a rather flat p_L as a function of λ, while in the batch-size based scheme p_L can vary quite significantly with λ. Our results indicate that in near-VOD, profit can be traded off with user loss: if the system has already achieved satisfactory profit, then the combined-for-loss scheme can be used to offer a better service quality; on the other hand, if the user loss rate can be relaxed, the combined-for-profit scheme can be used to achieve higher profit.

IV. Conclusion

In this paper, we have considered providing near VOD service when there is a cost associated with using a multicast channel. Batching then has to be done so as to amortize such channel cost so as to achieve profit, while meeting user delay or user loss (in the case of user reneging) requirement. We have examined and analyzed a number of (traditional) basic schemes, namely the window-based schemes and batch-size based scheme, in terms of their profit issues and user delay or loss. In general, there is a tradeoff between profit and user delay (or loss) in the system. The window-based schemes are able to offer guaranteed delay but not profit, while the batch-size based scheme is able to offer guaranteed profit but not delay. However, given a certain maximum user delay, batch-size based scheme is found to have lower profit when users do not renege. This makes the window-based schemes an attractive choice out of the basic schemes. We have also proposed and analyzed an adaptive scheme to combine the strengths of the basic schemes. The scheme is able to balance service quality (in terms of user delay experienced) and system profit when the underlying request rate fluctuates.

Given a certain user-reneging behavior, we find that maximizing profit per batch leads to long batching period and high user loss rate. Maximizing profit over a long time horizon (translated to the profit rate), on the other hand, is a better measure because it encourages more frequent smaller returns. However, the user loss rate may still be undesirably high at maximum profit, especially when the channel cost is high and users are not patient. Therefore, profit maximization should be subject to an acceptable level of user loss rate, and a shorter (suboptimal) batching period may have to be used. Generally, the higher the user delay tolerance is, the longer the batching period is and the higher the profit would be. Some unpopular movies would run at a revenue loss, unless incentives are used to make users to be more willing to wait or a higher PPV is charged. We finally show how the basic schemes can be combined to trade off profit and user loss in order to achieve high profit or low loss as the arrival rate fluctuates around some target or assumed value.

APPENDIX I
ANALYSIS OF THE ADAPTIVE SCHEME WHEN USERS DO NOT RENEGE

In this section, we analyze the adaptive scheme as presented in Section II-B. We first define a few variables. Let W be the random variable denoting the length of the batching period; clearly, $W_{\min} \leq W \leq W_{\max}$. Let α be the probability that $W = W_{\min}$, which is the case when more than $M-1$ users (excluding the first one in the batch) arrive within $W_{\min}$. Therefore

$$\begin{aligned}\alpha &= P(W = W_{\min})\\ &= 1 - \sum_{m=1}^{M-1} \frac{(\lambda W_{\min})^{m-1}}{(m-1)!}\, e^{-\lambda W_{\min}}.\end{aligned} \tag{14}$$

Let β be the probability that $W = W_{\max}$. It is equal to the probability that fewer than $(M-1)$ requests arrive (excluding the first request in the batch) within W_{max}, and is then given by

$$\begin{aligned}\beta &= P(W = W_{\max})\\ &= \sum_{m=1}^{M-1} \frac{(\lambda W_{\max})^{m-1}}{(m-1)!}\, e^{-\lambda W_{\max}}.\end{aligned} \tag{15}$$

Let γ be the probability that $W_{\min} < W < W_{\max}$. Clearly

$$\gamma = 1 - \alpha - \beta. \tag{16}$$

We first obtain the distribution of the batch-size, N, which is clearly given by (17), shown at the bottom of the page. The above equation can be used to find $\bar{N}$. Alternatively, $\bar{N}$ may be obtained by conditioning on W. Let $\bar{N}_\alpha = E[N \,|\, W = W_{\min}]$, $\bar{N}_\beta = E[N \,|\, W = W_{\max}]$, and $\bar{N}_\gamma = E[N \,|\, W_{\min} < W < W_{\max}]$. Clearly, $\bar{N} = \alpha\bar{N}_\alpha + \beta\bar{N}_\beta + \gamma\bar{N}_\gamma$, where

$$\bar{N}_\alpha = \sum_{m=M}^{\infty} m\, \frac{(\lambda W_{\min})^{m-1}}{\alpha(m-1)!}\, e^{-\lambda W_{\min}} \tag{18}$$

$$\bar{N}_\beta = \sum_{m=1}^{M-1} m\, \frac{(\lambda W_{\max})^{m-1}}{\beta(m-1)!}\, e^{-\lambda W_{\max}} \tag{19}$$

$$\bar{N}_\gamma = M. \tag{20}$$

We next obtain user delay distribution by conditioning on W. Given that $W = W_{\min}$ and there are m $(m \geq M)$ requests in the batch, one (the first one in the batch) has delay $W_{\min}$, and the remaining $(m-1)$ of them have delay uniformly distributed $\sim U[0, W_{\min}]$. Therefore, for $x \leq W_{\min}$, the user delay distribution is given by

$$\begin{aligned}f_D(x \,|\, W = W_{\min}) = \frac{1}{\bar{N}_\alpha} \sum_{m=M}^{\infty} \frac{(\lambda W_{\min})^{m-1}}{\alpha(m-1)!}\, e^{-\lambda W_{\min}}\\ \times \left((m-1)\frac{1}{W_{\min}} + \delta(x - W_{\min})\right)\end{aligned} \tag{21}$$

where $\delta(\cdot)$ is the usual impulse function. Similarly, given that a batching window is of size $W_{\max}$, the user delay distribution is $(x \leq W_{\max})$

$$\begin{aligned}f_D(x \,|\, W = W_{\max}) = \frac{1}{\bar{N}_\beta} \sum_{m=1}^{M-1} \frac{(\lambda W_{\max})^{m-1}}{\beta(m-1)!}\, e^{-\lambda W_{\max}}\\ \times \left((m-1)\frac{1}{W_{\max}} + \delta(x - W_{\max})\right).\end{aligned} \tag{22}$$

Given that $W = w$ with $W_{\min} < w < W_{\max}$ (and hence, the batch-size is exactly M), one user would have delay w, $(M-2)$ users would have delay $\sim U[0, w]$, and a user (the last one) would have zero delay. Therefore, for $0 \leq x \leq w$

$$\begin{aligned}&f_D(x \,|\, W_{\min} < w < W_{\max})\\ &\quad = \frac{1}{M}\delta(x - w) + \frac{M-2}{M}\frac{1}{w} + \frac{1}{M}\delta(x).\end{aligned} \tag{23}$$

$$P(N = i) = \begin{cases} \dfrac{(\lambda W_{\max})^{i-1}}{(i-1)!}\, e^{-\lambda W_{\max}}, & 1 \leq i \leq M-1\\[2ex] \dfrac{(\lambda W_{\min})^{M-1}}{(M-1)!}\, e^{-\lambda W_{\min}} + \gamma, & i = M\\[2ex] \dfrac{(\lambda W_{\min})^{i-1}}{(i-1)!}\, e^{-\lambda W_{\min}}, & i \geq M+1. \end{cases} \tag{17}$$

Removing the condition on w for $x \leq W_{\min}$, the delay distribution is given by

$$\begin{aligned} f_D(x) = \frac{1}{\bar{N}} \Bigg\{ & \beta \bar{N}_\beta f_D(x \,|\, W = W_{\max}) \\ & + \alpha \bar{N}_\alpha f_D(x \,|\, W = W_{\min}) \\ & + \gamma M \int_{W_{\min}}^{W_{\max}} f_D(x \,|\, W_{\min} < w < W_{\max}) \frac{g_W(w)}{\gamma} \, dw \Bigg\} \\ = \frac{1}{\bar{N}} \Bigg\{ & \sum_{m=1}^{M-1} \frac{(\lambda W_{\max})^{m-1}}{(m-1)!} e^{-\lambda W_{\max}} (m-1) \frac{1}{W_{\max}} \\ & + \sum_{m=M}^{\infty} \frac{(\lambda W_{\min})^{m-1}}{(m-1)!} e^{-\lambda W_{\min}} \left(\frac{m-1}{W_{\min}} + \delta(x - W_{\min}) \right) \\ & + (M-2) \int_{W_{\min}}^{W_{\max}} \frac{g_W(w)}{w} \, dw + \gamma \delta(x) \Bigg\} \end{aligned} \tag{24}$$

where $g_W(x)$ is given by (5). For $W_{\min} < x \leq W_{\max}$, the distribution is obtained similarly as

$$\begin{aligned} f_D(x) = \frac{1}{\bar{N}} \Bigg\{ & \sum_{m=M}^{\infty} \frac{(\lambda W_{\min})^{m-1}}{(m-1)!} e^{-\lambda W_{\min}} (m-1) \frac{1}{W_{\min}} \\ & + \sum_{m=1}^{M-1} \frac{(\lambda W_{\max})^{m-1}}{(m-1)!} e^{-\lambda W_{max}} \left(\frac{m-1}{W_{\max}} + \delta(x - W_{\max}) \right) \\ & + g_W(x) + (M-2) \int_{x}^{W_{\max}} \frac{g_W(w)}{w} \, dw \Bigg\}. \end{aligned} \tag{25}$$

$\bar{D}$ can then be obtained by $\int_0^{W_{\max}} x f_D(x) \, dx$.

Appendix II
Delay Distribution for Auto-Gating With User Reneging

We now derive user delay distribution for auto-gating with exponential user reneging function given by (8). For $W \leq U_{\min}$, users do not renege and $f_D(x)$ and $\bar{D}$ have already been derived in Section II-C. We now consider the case where $W > U_{\min}$. Note that the first user in a batch always has delay W. Requests arriving in the last $U_{\min}$ minutes of a batch never renege, and hence have delay uniformly distributed between $[0, U_{\min}]$. The remaining requests in the batch have delay $U_{\min} < x < W$ with probability $1 - R(x) = \exp(-(x - U_{\min})/\tau)$. Hence, the delay distribution $f_D(x)$ is given by

$$f_D(x) = \frac{1}{\bar{N}} \times \begin{cases} (\lambda U_{\min}) \dfrac{1}{U_{\min}} = \lambda, & \text{for } 0 \leq x \leq U_{\min} \\ \lambda e^{-(x - U_{\min})/\tau}, & \text{for } U_{\min} < x < W \\ \delta(x - W), & \text{otherwise.} \end{cases} \tag{26}$$

where $\bar{N}$ is the average batch-size given in (11). The average delay $\bar{D}$ can then be obtained by $\int_0^W x f_D(x) \, dx$.

Acknowledgment

The very early phase of this work was done during a summer job at the Hong Kong Telecom Institute of Information Technology in the Hong Kong University of Science and Technology, Hong Kong. The authors acknowledge the support of the group and, in particular, Dr. T.-M. Ko for providing the initial advice.

References

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

[2] V. O. K. Li and W. Liao, “Distributed multimedia systems,” *Proc. IEEE*, vol. 85, pp. 1063–1108, July 1997.

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

[4] K. C. Almeroth and M. H. Ammar, “The use of multicast delivery to provide a scalable and interactive video-on-demand service,” *IEEE J. Select. Areas Commun.*, vol. 14, pp. 1110–1122, Aug. 1996.

[5] W. Liao and V. O. K. Li, “The split and merge protocol for interactive video-on-demand,” *IEEE Multimedia Mag.*, pp. 51–62, Oct.–Dec. 1997.

[6] S.-H. G. Chan and E. Chang, “Providing scalable on-demand interactive video services by means of client buffering,” in *Proc. IEEE ICC*, Helsinki, Finland, June 2001.

[7] A. D. Gelman and S. Halfin, “Analysis of resource sharing in information providing services,” in *Proc. IEEE Globecom’90*, 1990, pp. 312–316.

[8] K. C. Almeroth, A. Dan, D. Sitaram, and W. H. Tetzlaff, “Long term resource allocation in video delivery systems,” in *Proc. IEEE INFOCOM’97*, Kobe, Japan, Apr. 1997, pp. 1333–1340.

[9] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, “On optimal batching policies for video-on-demand storage servers,” in *Proc. 3rd Int. Conf. Multimedia Computing and Systems*, Hiroshima, Japan, June 1996, pp. 253–258.

[10] A. Dan, D. Sitaram, and P. Shahabuddin, “Dynamic batching policies for an on-demand video server,” *ACM/Springer Multimedia Syst.*, vol. 4, pp. 112–121, June 1996.

[11] E. L. Abram-Profeta and K. Shin, “Scheduling video programs in near video-on-demand systems,” in *Proc. ACM Int. Multimedia Conf.*, Nov. 1997, pp. 359–371.

[12] S. Sheu, K. A. Hua, and W. Tavanapong, “Chaining: A generalized batching technique for video-on-demand systems,” in *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, Ottawa, ON, Canada, June 1997, pp. 110–117.

[13] L. Golubchik, J. C. S. Lui, and R. Muntz, “Reducing I/O demand in video-on-demand storage servers,” in *Proc. ACM SIGMETRICS’95*. Ottawa, ON, Canada, 1995, pp. 25–36.

[14] C. Aggarwal, J. L. Wolf, and P. S. Yu, “On optimal piggybacking merging policies for video-on-demand systems,” *Perform. Eval. Rev.*, vol. 24, pp. 200–209, May 1996.

[15] S. Viswanathan and T. Imielinski, “Metropolitan area video-on-demand service using pyramid broadcasting,” *Multimedia Syst.*, vol. 4, pp. 197–208, Aug. 1996.

[16] L.-S. Juhn and L.-M. Tseng, “Enhanced harmonic data broadcasting and receiving scheme for popular video service,” *IEEE Trans. Consumer Electron.*, vol. 44, pp. 343–346, May 1998.

[17] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, “Design and analysis of permutation-based pyramid broadcasting,” *ACM/Springer Multimedia Syst.*, vol. 7, no. 6, pp. 439–448, 1999.

[18] K. A. Hua and S. Sheu, “Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems,” *ACM Comput. Commun. Rev.*, vol. 27, pp. 89–100, Oct. 1997.

[19] L. Gao, J. Kurose, and D. Towsley, “Efficient schemes for broadcasting popular videos,” in *Proc. NOSSDAV’98*, Cambridge, U.K., July 1998.

[20] S.-H. G. Chan, F. A. Tobagi, and T.-M. Ko, “Providing on-demand video services using request batching,” *Proc. 1998 IEEE Int. Conf. Communications (ICC’98)*, pp. 1716–1722, June 1998.

[21] S.-H. G. Chan and F. A. Tobagi, “On achieving profit in providing near video-on-demand services,” in *Proc. 1999 IEEE Int. Conf. Communications (ICC’99)*, Vancouver, BC, Canada, June 1999, pp. 988–994.

[22] S.-H. G. Chan, “Scalable Services for Video-on-Demand,” Ph.D. dissertation, Dept. Elect. Eng., Stanford Univ., Stanford, CA, Jan. 1999.

S.-H. Gary Chan (M’91) received the B.S.E. degree (with highest honors) in electrical engineering from Princeton University, Princeton, NJ, in June 1993, and the Ph.D. degree in electrical engineering (with a minor in business administration) from Stanford University, Stanford, CA, in 1999, .

He is currently an Assistant Professor in the Department of Computer Science, Hong Kong University of Science and Technology (HKUST), Kowloon, and an Adjunct Researcher at Microsoft Research, Beijing, China. Prior to joining HKUST, he was a Visiting Assistant Professor in networking for a year with the University of California at Davis from September 1998 to June 1999. During 1992–1993, he was a research intern at the NEC Research Institute, Princeton, NJ. His research interests include multimedia networks, services, and systems, high-speed and wireless communications networks, and Internet technologies and protocols.

Dr. Chan was a William and Leila Fellow at Stanford University during 1993–1994 and the recipient of Charles Ira Young Memorial Tablet and Medal and the POEM Newport Award of Excellence in 1993 at Princeton University. He is a member of Tau Beta Pi, Sigma Xi, and Phi Beta Kappa.

Fouad A. Tobagi (M’77–SM’83–F’85) received the Engineering Degree from Ecole Centrale des Arts et Manufactures, Paris, France, in 1970 and the M.S. and Ph.D. degrees in computer science from the University of California at Los Angeles in 1971 and 1974, respectively.

From 1974 to 1978, he was a Research Staff Project Manager with the ARPA project at the Computer Science Department, University of California at Los Angeles, engaged in research in packet radio networks, including protocol design, and analysis and measurements of packet radio networks. In June 1978, he joined the faculty of the School of Engineering at Stanford University, Stanford, CA, where he is a Professor of Electrical Engineering and Computer Science. In 1991, he co-founded Starlight Networks, Inc., a venture concerned with multimedia networking, and served as Chief Technical Officer until August 1998. He was co-editor of the book *Advances in Local Area Networks* (New York: IEEE Press) in the Frontiers in Communications series. His research interests include packet switching in ground radio and satellite networks, high-speed local area networks, fast packet switching, broadband integrated-services digital networks, asynchronous transfer mode, multimedia networking and communications, and modeling and performance evaluation of network systems.

Dr. Tobagi served as Associate Editor for Computer Communications in the IEEE TRANSACTIONS ON COMMUNICATIONS during 1984–1986, Editor for Packet Radio and Satellite Networks in the *Journal of Telecommunications Networks* during 1981–1985, Co-Editor of the Special Issue on Local Area Networks of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (November 1983), Co-Editor of the Special Issue on Packet Radio Networks of the PROCEEDINGS OF THE IEEE (January 1987), and Co-Editor of the Special Issue on Large Scale ATM Switching Systems for B-ISDN of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (October 1991). He is currently serving as Editor for a number of journals on high-speed networks, wireless networks, multimedia, and optical communications. He is a member of the Association for Computing Machinery (ACM) and has served as an ACM National Lecturer during 1982–1983. He is co-recipient of the 1998 Kuwait Prize in the field of Applied Sciences, the winner of the 1981 Leonard G. Abraham Prize Paper Award in the field of communications systems for his paper “Multiaccess Protocols in Packet Communications Networks,” and co-winner of the IEEE Communications Society 1984 Magazine Prize Paper Award for the paper “Packet Radio and Satellite Networks.”