Providing On-Demand Video Services Using Request Batching

Shueng-Han Gary Chan Fouad Tobagi
Department of Electrical Engineering
Stanford University
Stanford, CA 94305

Tsz-Mei Ko
Department of Electrical and Electronic Engineering
The Hong Kong University of Science and Technology
Kowloon, Hong Kong

Abstract— **In an on-demand video system, requests for a video file arriving within a period of time can be batched together and served with a single multicast stream; thereby reducing the bandwidth requirement compared with the unbatched case. In this paper, we study various batching schemes to meet delay and profitability requirements — in the window-size based and moving-average schemes, maximum and average user delay respectively are guaranteed; in the batch-size based scheme, minimal profitability is maintained; in the adaptive scheme, both the delay and profitability can be balanced. We analyse and compare these schemes in terms of the delay experienced by the users, and their profitability (the number of users in a batch, the number of concurrent streams, etc.).**

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[4]. Hence users enjoy great flexibility in interacting with the server viewing their videos [5]. However, pure-VOD does not scale up well with the user population and becomes very expensive when a large number of concurrent requests have to be accommodated.

When the video content is popular and user interactivity is not required, as is the case with some applications such as movie-on-demand, 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 arriving over a period of time and serving these requests by a single multicast stream. This is referred to as near-VOD [6], [7], [8]. However, the bandwidth saving as compared to pure-VOD is achieved at the expense of user start-up delay. Today, this technique is used in many satellite and cable-based movie-on-demand services. It basically consists of having the same movie shown at specific pre-scheduled 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 size is, the smaller is the bandwidth used and the larger the batch size is (which, in turn, translate to higher revenue per stream), but the longer is the delay experienced by a user.

In this paper, we consider a number of schemes for batching user requests. These schemes are analyzed and compared with each other in terms of the delay experienced by a user and the system profitability (i.e., the number of concurrent streams and the number of users per streams).

Dan et. al., in 1996 considered the problem of scheduling the outstanding requests for a near video-on-demand system when bandwidth is extremely limited. They studied the system throughput given certain user reneging behavior [6]. Our focus in this paper is on a single movie under the assumptions that: i) there is no constraint on the available streams that could be used for the movie, but there is a cost associated with the use of these streams (Issues associated with insufficient bandwidth are addressed in [9]); ii) users waiting for their requests to be served do not renege (particularly justified if a certain user delay requirement is satisfied).

The paper is organized as follows. We first describe in Sect. II the batching schemes we consider in the paper. In Sect. III, we present an analysis of these schemes and their performance characteristics in terms of the number of concurrent streams, the number of users served by a stream, and the user delay experienced under the assumption that the arrival process is Poisson. In Sect. IV, we present and discuss some numerical results as well as the comparison among the various schemes. We conclude in Sect. V.

II. System Description

We consider four classes of schemes: Window-based schemes, batch-size based scheme, adaptive scheme and moving-average scheme

A. Window-size based schemes

Consider a movie of duration T_h minutes. The *fixed scheduling* scheme is the simplest of all batching schemes whereby a movie is shown once every exactly W minutes ($W \leq T_h$). The number of required concurrent multicast streams is deterministic and is simply given by $\lceil T_h/W \rceil$. A user which makes a request to view the video between two such showings (the batching window) is served by the next showing following the request. If the average rate of arrival of requests is λ requests/minute (60λ req./hr.), then the average number of users served by a stream is simply λW.

In the fixed scheduling, a stream is used even if no request has been made in the batching period proceeding it. It is possible to omit a showing if no user has requested the movie in its corresponding batching window. This scheme is referred to as *fixed gating.* The number of concurrent streams is not deterministic any more, but is a random variable which depends on the arrival process. In both fixed scheduling and fixed gating batching schemes, the user delay is bounded by W. Furthermore, the show time may be published ahead of time.

A third batching scheme guaranteeing a delay bound of W is that the batching window of size W following the start of a showing is started with the first arrival following the start of that showing. We refer to this scheme as the *auto-gated scheduling.* While in this scheme, the show time cannot be announced ahead of time, each

This work was supported by the Center for Telecommunications at Stanford University, and a grant from Hong Kong Telecom Institute of Information Technology at the Hong Kong University of Science and Technology.

such show time is determined and may be advertised as soon as the start of the corresponding batching window is determined.

B. Batch-size based scheme

In general, the concern of service providers is to have as many users in a batch as possible, since the system profitability is directly related to the average batch size. Let C_s be the cost in using a stream for the display of a movie, and let $\overline{N}$ be the average number of users in a batch (the average batch size). If P is the price charged to a user for viewing the video, then on average each stream would bring in a revenue equal to $\overline{N}P$. To maintain profitability, we need $\overline{N}P \geq C_s$; defining $K \triangleq C_s/P$, we need $\overline{N} \geq K$. The higher the batch size, the more profitable the system is.

Note that in auto-gating, the number of users in a batch is high when the arrival rate is high. However, when the arrival rate is low, system profitability suffers. In batch-size based scheme, we secure 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 in such a scheme, users suffer uncertain waiting time in the system.

C. Adaptive scheme

Given the remarks made above, we wish to operate according to a window-size based scheme when the rate of arrivals is high enough to achieve profitability, and according to the batch-size based scheme when the arrival rate drops to maintain profitability. Therefore we propose the following self-adaptive scheme which achieves this goal. We consider that the user satisfaction is high if the delay the user experienced is below a certain value D_{min}. We also consider that there is a delay $D_{max} > D_{min}$ beyond which the user satisfaction is low and, in this case, the probability of user reneging is high. Delays between D_{min} and D_{max} are tolerated and within this range no reneging is likely, 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 operate as follows: 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 batched, 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 batch-size based scheme with $M \geq K$; but since users may renege if they wait higher than D_{max}, a maximum batching window of W_{max} is imposed (even if there are fewer than M users arriving within the window of size W_{max}, the movie is shown anyway). Users, in this system, 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.

D. Moving-average scheme

This scheme addresses the objective of satisfying a certain average user delay requirement $\hat{D}$. Such a requirement may be met by dynamically adjusting the batching window so that the mean delay of the users in each batch is equal to $\hat{D}$. Given a set of known request arrival times $A_1, A_2, \ldots, A_i$ in a batch (counted from the time of the first arrival in the batch, and hence $A_1 = 0$), in order to keep the average delay in the batch equal to $\hat{D}$, the (future) tentative display time T_D has to satisfy $(1/i)\sum_{j=1}^{i}(T_D - A_j) = \hat{D}$, which gives,

$$T_D = \hat{D} + \frac{\sum_{j=1}^{i} A_j}{i}, \tag{1}$$

i.e., the video display time is the average delay requirement $\hat{D}$ plus the average arrival time in the batch. Therefore the show time of a video has to be re-calculated at each arrival in a batch.

III. Analysis

In this section, we analyze the performance of the batching schemes mentioned above. We focus on a particular video in the system, with the request's arrival process being Poisson with rate λ. In evaluating the various schemes described in Sect. II, we are interested in the following closely-related performance measures:

• *The required number of concurrent streams $S(t)$, both its distribution $f_S(s)$ and its mean value $\overline{S}$:* Average number of concurrent streams indicates the average bandwidth required in the system. We may also be interested in the bandwidth saving as compared to pure-VOD; since the average number of concurrent streams required in pure-VOD is λT_h, the bandwidth saving η is defined as $\eta = 1 - \overline{S}/(\lambda T_h)$.

• *The batch size N, both its distribution $f_N(n)$ and its mean $\overline{N}$:* Note that $\overline{N}$ is related to $\overline{S}$ by $\lambda T_h = \overline{S}\ \overline{N}$. Indeed, by the Little's formula, the average number of user requests which arrive during the video display time T_h is simply λT_h, which is also equal to $\overline{S}\ \overline{N}$.

• *The delay D incurred by a random user, both its distribution $f_D(t)$ and its mean $\overline{D}$.*

A. Window-based schemes

In fixed scheduling, obviously, the distribution of the batch size is Poisson with mean $\overline{N} = \lambda W$, and since a video is shown exactly once every W minutes, $\overline{S} = T_h/W$ and $\eta = 1 - 1/(\lambda W)$. D is uniformly distributed $\sim U[0, W]$ and $\overline{D} = W/2$.

In fixed gating, the probability that there are i ($i \geq 1$) requests in a batch is given by $(\lambda W)^i e^{-\lambda W}/(i!(1 - e^{-\lambda W}))$; hence, $\overline{N} = \lambda W/(1 - e^{-\lambda W})$, and

$$\overline{S} = \frac{T_h}{W}(1 - e^{-\lambda W}). \tag{2}$$

As expected, when $\lambda \to 0$, $\overline{S} \to \lambda T_h$ (the number corresponding to pure-VOD); and $\overline{S} \leq T_h/W$ with $\lim_{\lambda\to\infty} \overline{S} = T_h/W$. The bandwidth saving is therefore, $\eta = 1 - (1 - e^{-\lambda W})(\lambda W)$.

Given that there is an arrival within a batching window W, its arrival time is uniformly distributed within the window. Hence, the user delay distribution for the video is, $f_D(t) = 1/W$, for $0 \leq t \leq W$, and $\overline{D} = W/2$.

For the auto-gated scheduling, The distribution of the batch size is $P(N = i) = (\lambda W)^{i-1} e^{-\lambda W}/(i-1)!$, for $i \geq 1$. Hence $\overline{N} = 1+\lambda W$ (which is exactly one more than that of fixed scheduling), and

$$\overline{S} = \frac{\lambda T_h}{1 + \lambda W}, \tag{3}$$

and $\eta = \lambda W/(1 + \lambda W)$.

The delay distribution is given by,

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

where $\delta(t)$ is the usual impulse function with $\delta(t) = 0$ for $t \neq 0$ and $\int_{-\infty}^{\infty} \delta(t)dt = 1$. It accounts for the first user in each batch having a delay of W while the remaining users in the batch having a delay uniformly distributed between 0 and W. $\overline{D}$ is hence given by,

$$\overline{D} = \frac{\lambda W + 2}{2(1 + \lambda W)}W. \tag{5}$$

B. Batch-size based scheme

The movie is displayed each time a new batch of M requests is collected. Therefore, N is deterministic and equal to M; hence $\overline{N} = M$, $\overline{S} = \lambda T_h/M$, and $\eta = 1 - 1/M$.

Let W denote the batching period. It is a random variable equal to the sum of $(M-1)$ exponential variables, and its distribution $g_W(w)$ is given by,

$$g_W(w) = \frac{\lambda(\lambda w)^{M-2}}{(M-2)!} e^{-\lambda w}. \tag{6}$$

The user delay distribution may be obtained by conditioning on W, and is given by [10],

$$f_D(x) = \tfrac{1}{M} g_W(x) + \tfrac{M-2}{M} \int_x^{\infty} \tfrac{g_W(w)}{w} dw + \tfrac{\delta(x)}{M}, \tag{7}$$

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

C. Adaptive scheme

The distribution of N, which can be used to find $\overline{N}$, is given by,

$$P(N=i) = \begin{cases} \frac{(\lambda W_{max})^{i-1}}{(i-1)!} e^{-\lambda W_{max}} & \text{for } 1 \le i \le M-1, \\ \frac{(\lambda W_{min})^{M-1}}{(M-1)!} e^{-\lambda W_{min}} + \gamma & \text{for } i = M, \\ \frac{(\lambda W_{min})^{i-1}}{(i-1)!} e^{-\lambda W_{min}} & \text{for } i \ge M+1, \end{cases} \tag{8}$$

where $\gamma = \int_{W_{min}}^{W_{max}} g_W(w)dw$ ($g_W(w)$ is given in Eq. (6)).

We get the user delay distribution by conditioning on W [10]. For $x \le W_{min}$, the delay distribution is given by

$$\begin{aligned} f_D(x) &= \frac{1}{\overline{N}} \Bigg\{ \sum_{m=1}^{M-1} \frac{(\lambda W_{max})^{m-1}}{(m-1)!} e^{-\lambda W_{max}} (m-1) \frac{1}{W_{max}} + \\ &\quad \sum_{m=M}^{\infty} \frac{(\lambda W_{min})^{m-1}}{(m-1)!} e^{-\lambda W_{min}} \left(\frac{m-1}{W_{min}} + \delta(x - W_{min}) \right) \\ &\quad + (M-2) \int_{W_{min}}^{W_{max}} \frac{g_W(w)}{w} dw + \gamma \delta(x) \Bigg\}, \end{aligned} \tag{9}$$

and for $W_{min} < x \le W_{max}$, the distribution is,

$$\begin{aligned} f_D(x) &= \frac{1}{\overline{N}} \Bigg\{ \sum_{m=M}^{\infty} \frac{(\lambda W_{min})^{m-1}}{(m-1)!} e^{-\lambda W_{min}} (m-1) \frac{1}{W_{min}} + \\ &\quad \sum_{m=1}^{M-1} \frac{(\lambda W_{max})^{m-1}}{(m-1)!} e^{-\lambda W_{max}} \left(\frac{m-1}{W_{max}} + \delta(x - W_{max}) \right) \\ &\quad + g_W(x) + (M-2) \int_x^{W_{max}} \frac{g_W(w)}{w} dw \Bigg\}. \end{aligned} \tag{10}$$

$\overline{D}$ may then be obtained by $\int_0^{W_{max}} x f_D(x) dx$.

D. Moving-average scheme

The analysis for this scheme has proven to be rather difficult. Accordingly, we have used simulation to study its performance.

IV. Numerical Results and Comparisons

A. Window-based schemes

The auto-gated scheduling is chosen as a representative of window-based schemes. We consider $T_h = 90$ minutes. In Fig. 1, we plot $\overline{S}$ versus W, given λ. Also shown in the figure is the maximum T_h/W which is attained when $\lambda \to \infty$, and corresponds to the number required in the fixed scheduling case. In Fig. 2, we plot $\overline{S}$ versus λ, given W. Given λ, when $W = 0$, $\overline{S} = \lambda T_h$ corresponds to pure-VOD. As W increases, $\overline{S}$ decreases from λT_h, first rapidly and then slowly; most of the bandwidth reduction as compared to pure-VOD is attained for a relatively low value of W, in the range of 6–8 minutes, and beyond this range the rate of reduction becomes relatively small. Furthermore, this gain is significantly more important for larger λ than for smaller λ. We also note from both figures that the bandwidth reduction as compared to the fixed schedule (T_h/W) is more significant for smaller W and smaller λ. Indeed, the average number of streams which can be expressed as $T_h/(W + 1/\lambda)$ is small compared to T_h/W when $(W + 1/\lambda)$ is large as compared to W; this is the case when W is small given λ, and it is more pronounced when $1/\lambda$ is large.

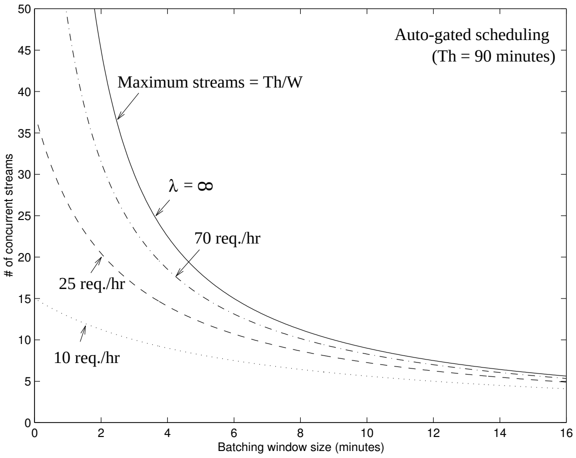


Fig. 1. $\overline{S}$ versus W for the auto-gated scheduling

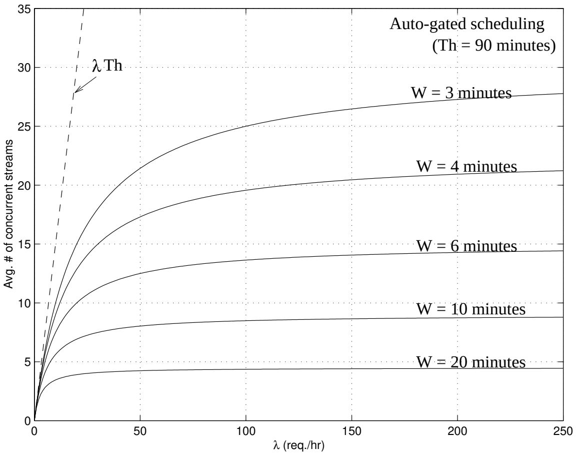


Fig. 2. $\overline{S}$ versus λ for the auto-gated scheduling

We plot in Fig. 3 $\overline{D}$ versus W for given λ (the delay distribution for various schemes have been examined in [10]). From Eq. (5), $\overline{D}$ can be re-written as $\overline{D} = W/2 + (1/2)/(1/W + \lambda) \le W/2 + 1/(2\lambda)$. The upper bound is reached when W gets large so as $1/W$ gets small as compared to λ; for large λ, this is the case for even small value of W (e.g., for $\lambda = 120$ req./hr, $W \ge 5$ minutes).

In Fig. 4, we plot $\overline{D}$ versus λ, given W. The user delay at $\lambda = 0$ is equal to W with probability 1 (each batch would only have one user). As λ increases, $\overline{D}$ decreases from W, reaching asymptotically $W/2$. We observe that the drop in $\overline{D}$ as λ increases

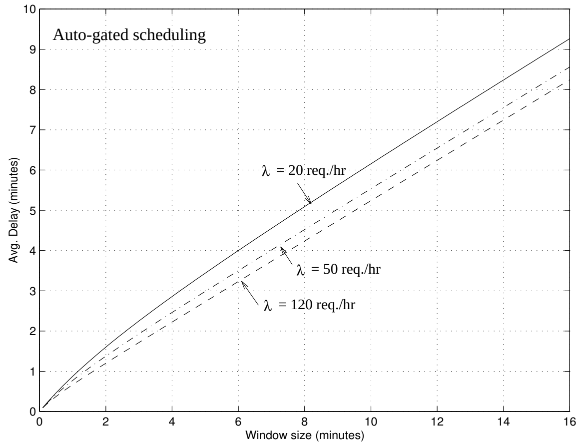


Fig. 3. Average delay as a function of W given λ

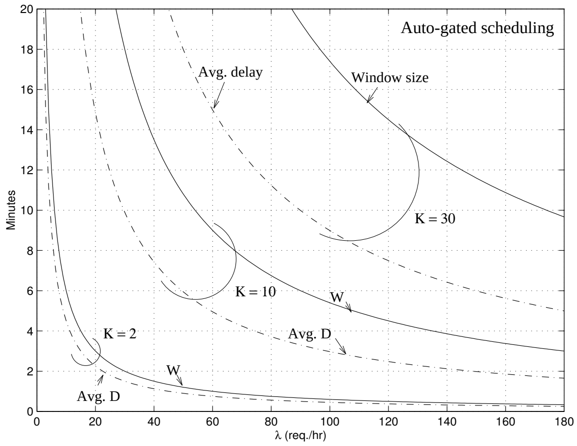


Fig. 5. W and the corresponding $\overline{D}$ versus λ given certain K

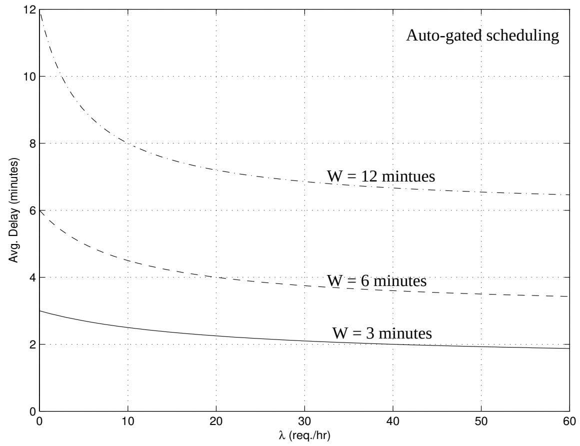


Fig. 4. $\overline{D}$ as a function of λ for the auto-gated scheme

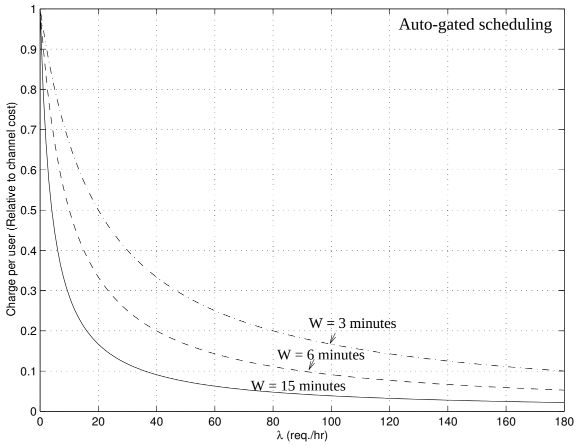


Fig. 6. Charge per user (normalized to C_s) versus λ

is sharper with larger values of W than with smaller value of W. This is expected since more users can be batched with a larger window. It is encouraging to note that for window size of only 12 minutes, the users experience an average delay of 8 minutes for an arrival rate as low as 10 req./hr., and an average delay close to 6 minutes for an arrival rate equals to 40 req./hr.

We consider now the auto-gated scheme in which W is selected appropriately for each value of λ so as to guarantee a given average batch size M, and thus a given level of profitability (where M is selected to be larger than K). (Clearly, this assumes that the rate λ of the arrival process is known *a priori* and remains constant over relatively long periods of time.) In Fig. 5, we plot the minimum window size W necessary to guarantee $1 + \lambda W = K$, and the resulting average delay incurred, as a function of λ. Clearly, for a given K, larger values of λ require smaller values of W, and the higher K is, the larger must W be. For $\lambda = 40$ req./hr and $K = 10$, W has to be 14 minutes in order to maintain profitability; and if K is now 2 as opposed to 10, W may be decreased to 1.5 minutes.

One may achieve both profitability and a maximum delay constraints if one were to charge users a fee per movie P appropriately chosen as a function of λ. P should be such that $K \equiv C_s/P = 1 + \lambda W$, and thus the fee normalized to C_s is given by $P/C_s = 1/(1+\lambda W)$. We plot in Fig. 6 P/C_s versus λ for given W. For very low arrival rates, when there is most likely only 1 user in a batch, the fee to be charged is equal to the stream cost. As λ increases, the fee to be charged so as to break even drops quickly; for example, for $W = 6$ minutes when $\lambda = 40$ req./hr, the charge needs to be $0.2C_s$.

We now compare pure-VOD, fixed scheduling, fixed gating and auto-gated scheduling in terms of $\overline{S}$ and $\overline{N}$ for given values of λ and W. Since in all cases $\overline{S}$ is expressed in terms of the product λW, we plot in Fig. 7 $\overline{S}$ normalized to T_h/W as a function of λW. Obviously, the auto-gated scheme consumes less bandwidth than fixed gating. In fact (by using Eqs. (2) and (3)), it is easy 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 $\overline{S}$ between the two is $0.2036T_h/W$ (attained at $\lambda W = 2.51276$). Fixed scheduling always consumes on average T_h/W streams no matter what the arrival rate is, and both the fixed gating and the auto-gated schemes consume less bandwidth than fixed scheduling. However, as λW gets large, fixed scheduling is as good as fixed gating, and ultimately, the auto-gated scheme will asymptotically reach the same value. Note also that for $\lambda \leq 1/W$, even pure-VOD performs better than fixed scheduling.

We plot in Fig. 8 $\overline{N}$ for the various window-based schemes as a function of λ. We note that the batch sizes for all window-based schemes differ by at most one.

With respect to delay, for the same W, all window-based schemes guarantee the same maximum delay W. The average delay however is fixed at $W/2$ for both fixed scheduling and fixed gating, while that of the auto-gated scheme is between $W/2$ and

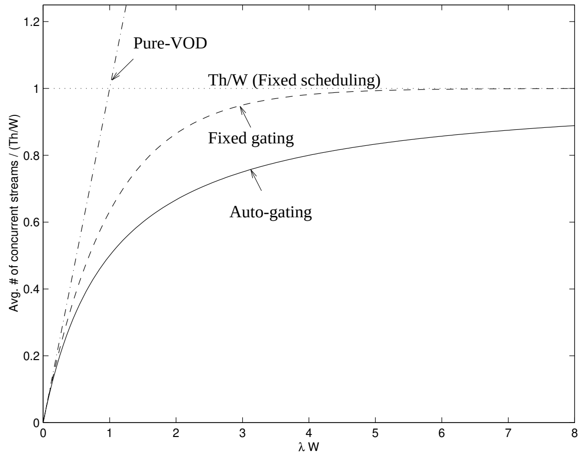


Fig. 7. $\overline{S}$ (normalized to T_h/W) versus λW

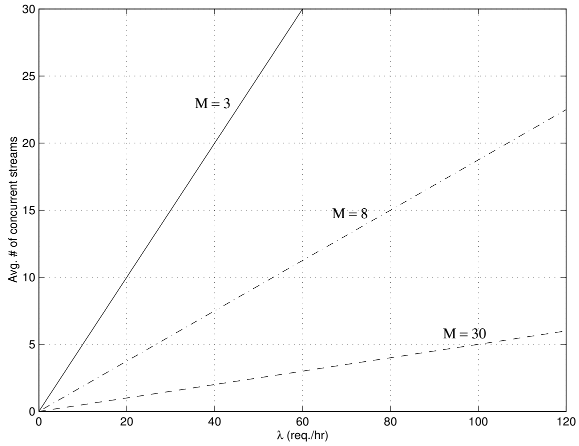


Fig. 9. $\overline{S}$ versus λ given M

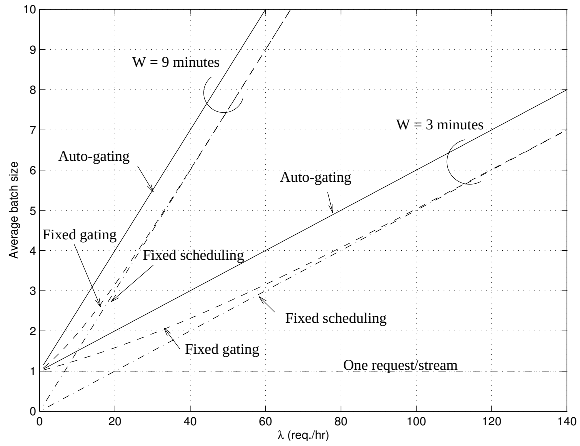


Fig. 8. $\overline{N}$ versus λ given W

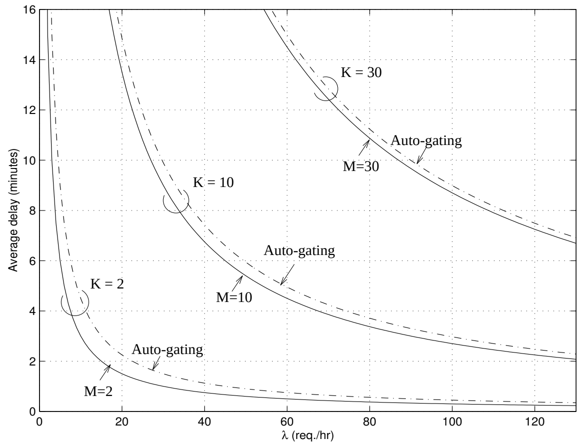


Fig. 10. $\overline{D}$ versus λ given K

W depending on the intensity of λ (see Fig. 4).

B. Batch-size based scheme

The batch-size based scheme is motivated by the need to guarantee a certain number of users per stream M (so as to guarantee a certain level of profitability). Each batching period is extended until exactly M users have been accumulated.

$\overline{S}$ versus λ given M is plotted in Fig. 9. Clearly, $\overline{S}$ is linear in λ, and in contrast with what we have seen for window-based schemes in Sect. IV-A, $\overline{S}$ keeps increasing as λ increases, and does not settle to a limiting value. Therefore, if M is to remain fixed, when λ gets large, the batch-size based scheme in fact consumes more bandwidth than window-based schemes.

We now consider the design of auto-gated and batch-size based systems so as to satisfy a given profitability K and compare their average delay. We plot in Fig. 10 the average delay as a function of λ for given K. We see that both schemes have average delays that are fairly close (with the batch-size based scheme having slightly lower delay).

C. Adaptive scheme

We plot in Fig. 11 $\overline{S}$ versus λ. We also show (in light dotted lines) the corresponding $\overline{S}$ for the batch-size based scheme (with M set to the value used in the adaptive scheme) and for the auto-gated scheme (with W set to the various values of W_{min} and W_{max} used in the adaptive scheme). We see that as λ increases, the average number of concurrent streams first behaves as in the auto-gated scheme with window size W_{max}, then it switches over to the batch-size based scheme (shown as the straight line), and then finally, as the arrival rate increases further, it behaves as in the auto-gated scheme with window size W_{min}.

In Fig. 12, we show the average batch size $\overline{N}$ as a function of λ; we also show in light dotted lines the same curves for the auto-gated scheme with the corresponding parameters. Similar to $\overline{S}$, we see that as λ increases, $\overline{N}$ follows the auto-gated scheme curve with window size W_{max}, and then the batch-size plateau with the corresponding parameter, and finally the auto-gated scheme again with window size W_{min}. Note that the portion of the graph lying above $N = M$ corresponds to the profitable region. For a time varying arrival rate, the total system profitability depends on how often the process visits the upper portion (with $N \geq M$) and the lower portion.

In Fig. 13, we plot the avererage delay as a function of λ. We see that when the arrival rate is low, the average delay is that of auto-gated scheme with window size W_{max}, and as the arrival rate increases, it switches to that of auto-gated scheme with window size W_{min}.

D. Moving-average scheme

In the moving-average scheme, the movie show-time is pushed forward in time each time there is a new arrival in the batch. Therefore we are interested in the user delay distribution given

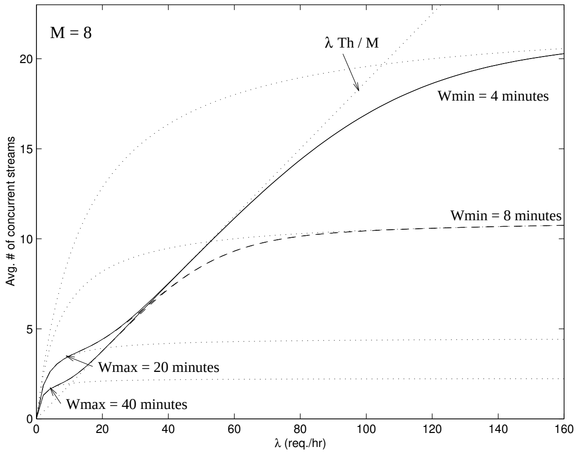


Fig. 11. $\overline{S}$ versus λ for the adaptive scheme

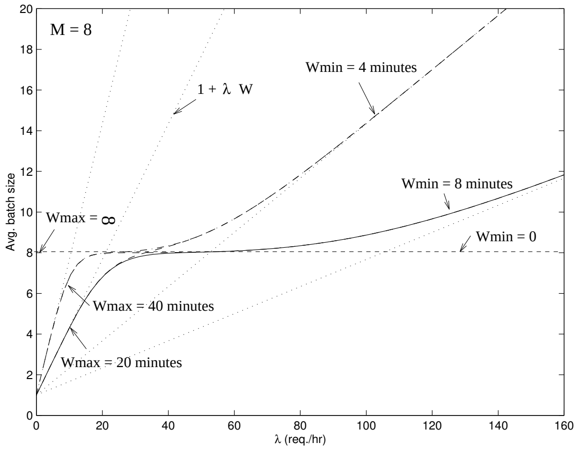


Fig. 12. $\overline{N}$ versus λ for the adaptive scheme

that a certain average delay requirement $\hat{D}$ is to be satisfied. In Figs. 14(a) and 14(b), we show the user delay distribution with low arrival rate ($\lambda = 10$ req./hr) and high arrival rate ($\lambda = 120$ req./hr), respectively, for $\hat{D} = 1$ minute, 3 minutes, and 10 minutes. We see that most users actually enjoy a delay equal to the averages, and almost no user suffers a delay higher than $2\hat{D}$. It is encouraging to see that even though the batching window keeps lengthening with each user arrival, users are not delayed indefinitely. In fact, as λ increases, the average arrival time approaches the mid-point between the first and the last arrival of a batch (the batching period). Therefore, with the use of Eq. (1), we have $\lim_{\lambda\to\infty} W = 2\hat{D}$, and the delay distribution approaches $\sim U[0, 2\hat{D}]$. This is actually verified when $\lambda = 120$ req./hr. Hence, as the arrival rate increases, $\overline{S}$ settles to a limiting value of $T_h/(2\hat{D})$.

In Fig. 15, we show $\overline{S}$ versus λ given $\hat{D}$. As expected, as λ increases, $\overline{S}$ increases and approaches the limiting values $T_h/(2\hat{D})$. For an average delay of as low as 5 minutes, only 9 streams is sufficient to serve hundreds of requests per hour.

In Fig. 16, we compare $\overline{S}$ as a function of λ for the moving-average scheme, auto-gated and batch-size based schemes, given a certain average delay requirement ($\hat{D} = 3$ minutes). We see that the batch-size based scheme consumes the least bandwidth among all the schemes. In fact (from Eqs. (3), (5) and $M = 1+2\hat{D}\lambda$), it is not hard to show that auto-gated scheduling consumes up to 25% (at $\lambda\hat{D} = 0.75$) more bandwidth than the batch-size based scheme,

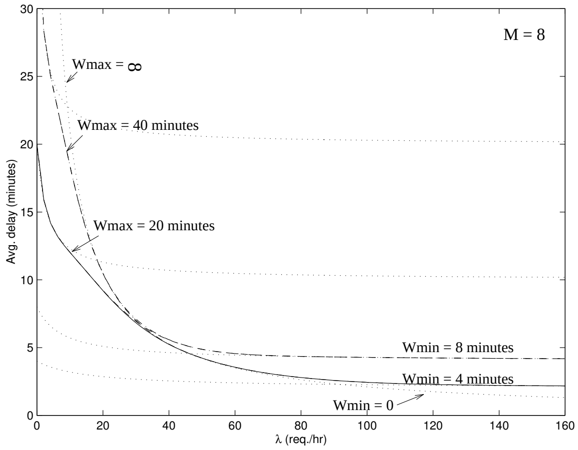


Fig. 13. Average delay versus λ for the adaptive scheme

while the maximum difference in $\overline{S}$ between the batch-size based scheme and the auto-gated scheme is $0.082T_h/\hat{D}$ (at $\lambda = 1.15/\hat{D}$). Note that in all schemes, the limiting value of $\overline{S}$ as λ increases is $T_h/(2\hat{D})$. We see that only 15 streams is needed to provide $\hat{D} =$ 3 minutes.

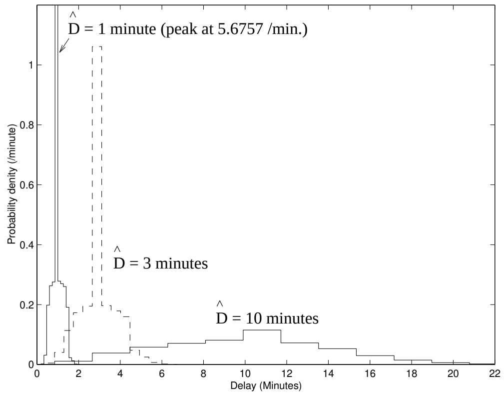
In Fig. 17, we compare $\overline{N}$ as a function of λ given a certain $\hat{D}$. We see that there is not much difference in the batch sizes among all the schemes.

V. Conclusions

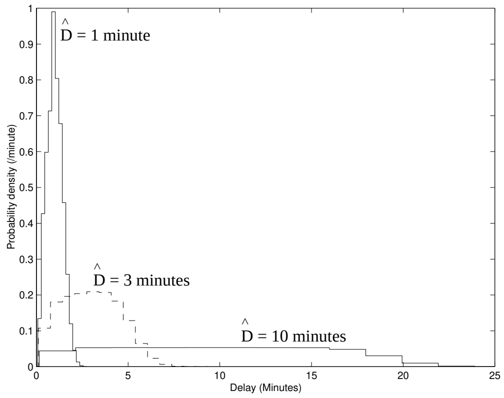
Video services in which requests for a particular video are batched together and served with a single I/O stream can potentially save bandwidth tremendously. In this paper we consider a number of ways to do such batching so as to meet requirements such as user delay and system profitability.

We have studied window-based schemes in which users are batched for a fixed period of time W before they are served, and hence user delay is bounded by a certain maximum value W. Since system profitability depends on the average number of users in a batch, window-based schemes has good profit when the arrival rate is high, but not so when the arrival rate decreases. Therefore we have studied another batch-size based scheme in which we maintain the profitability by opening a stream when a certain number M of users are collected in a batch. However, we found that the average number of streams increases without limit (while average user delay decreases to zero) as request rate increases. We therefore proposed a self-adaptive scheme with 3 parameters: i) a minimum window size W_{min} in which we batch as many users as possible when the arrival rate is high, ii) a batch-size M selected to guarantee profitability when the arrival rate is not so high, iii) a maximum window size W_{max} to bound user delay to W_{max} when the arrival rate is low. Depending on the arrival rate, the system operates as in one scheme or the other. Finally, we considered a batching scheme in order to meet a certain average delay requirement, in which we force the mean delay of all users in a batch to be equal to the requirement.

These schemes have been analyzed and compared with each other in terms of delay experienced by users and their profitability (the number of concurrent streams, the number of users in a batch, etc.), under the assumption of Poisson arrival. We found that the average number of concurrent streams in all the schemes settles to a low limiting number of streams as arrival increases, even with the average user delay of only a few minutes. Hence, request batching is able to provide video services to large number



a) $\lambda = 10$ req./hr



b) $\lambda = 120$ req./hr

Fig. 14. Delay distribution for the moving-average scheme

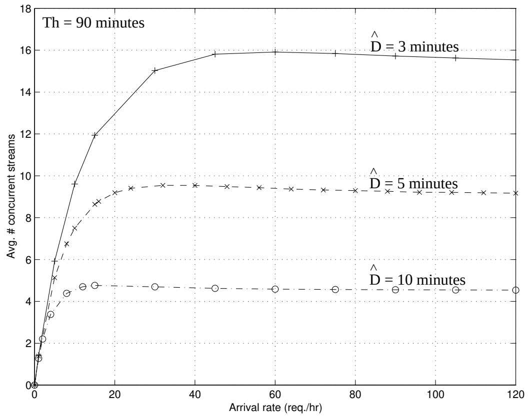


Fig. 15. $\overline{S}$ versus λ given $\hat{D}$ for the moving-average scheme

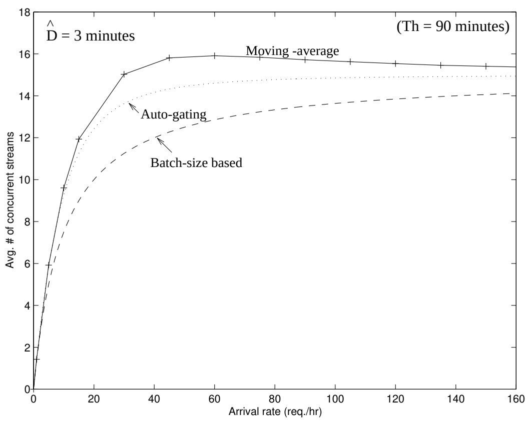


Fig. 16. $\overline{S}$ versus λ

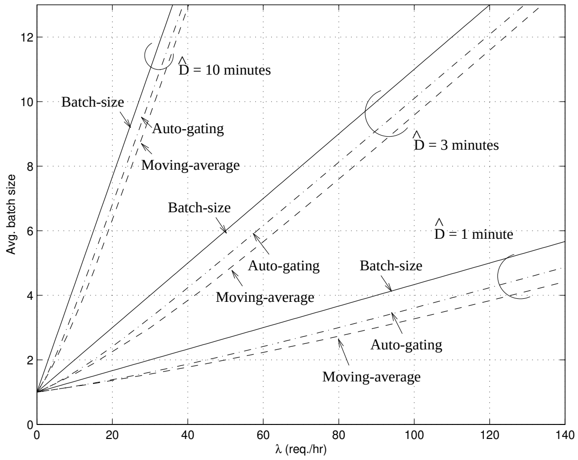


Fig. 17. $\overline{N}$ versus λ given $\hat{D}$

of concurrent users cost-effectively.

References

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

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

[3] Borko Furht, Deven Kalra, Frederick L. Kitson, Arturo A. Rodriguez, and William E. Wall, "Design issues for interactive television systems," *IEEE Computer Magazine*, pp. 25–38, May 1995.

[4] K. Cleary, "Video on demand — competing technologies and services," in *Proceedings of International Broadcasting Convention*, Amsterdam, Netherlands, 14–18 Sept. 1995, pp. 432 – 437.

[5] Fouad A. Tobagi, Joseph Pang, Randall Baird, and Mark Gang, "Streaming RAID$^{\mathrm{TM}}$ – a disk array management system for video files," in *Proceedings of First ACM International Conference on Multimedia*, Anaheim, CA, August 1-6 1993.

[6] A. Dan, D. Sitaram, and P. Shahabuddin, "Dynamic batching policies for an on-demand video server," *Multimedia Systems*, vol. 4, no. 3, pp. 112 – 121, June 1996.

[7] Shenze Chen and Manu Thapar, "A novel video layout strategy for near-video-on-demand servers," in *Proceedings of IEEE International Conference on Multimedia Computing and Systems'97*, Ottawa, Ont., Canada, 3–6 June 1997, pp. 37–45.

[8] Victor O. K. Li, Wanjiun Liao, Xiaoxin Qiu, and Eric W. M. Wong, "Performance model of interactive video-on-demand systems," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 6, pp. 1099–1109, August 1996.

[9] Shueng-Han Gary Chan, Fouad Tobagi, and Tsz-Mei Ko, "Auto-gated scheduling for near video-on-demand," to appear in *Proceedings of ISCAS'98*, CA, 1998.

[10] Shueng-Han Gary Chan, Fouad Tobagi, and Tsz-Mei Ko, "Providing on-demand video services using request batching," technical report (in preparation), Computer System Laboratory, Stanford University, CA.