BANDWIDTH PLANNING IN NEAR VIDEO-ON-DEMAND

S.-H. G. Chan, F. Tobagi

Dept. of Electrical Engineering
Stanford University
Stanford, CA 94305
USA

T.-M. Ko

Dept. of Electric. & Electron. Eng.
The Hong Kong University of
Science and Technology
Kowloon, Hong Kong

ABSTRACT

In near video-on-demand, requests for a video file arriving within a period of time are batched together and served with a single multicast stream; thereby reducing the bandwidth requirement compared with the unbatched case. Since the delayed users may renege (and hence causing revenue loss to the service providers), the design of a near video-on-demand system has to meet users' delay expectation. In this paper, we specify the minimum number of streams required for a video file so that only a small fraction of the users experience delay higher than a certain value. Trade-off issues between user's delay tolerance and the required number of streams are also addressed.

1. 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]

In pure-VOD, each user is assigned its own dedicated unicast stream[3]. Hence users enjoy great flexibility in interacting with the server while viewing their videos [4]. 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) the requests for a given movie arriving over a period of time and serving these requests by a single multicast stream. This is referred to as near-VOD [5, 6, 7, 8]. Today, this technique is used in many satellite and cable-based movie-on-demand services.

Note that the bandwidth saving in near-VOD as compared to pure-VOD is achieved at the expense of user start-up delay. Since user may renege while waiting to be served (and hence causing revenue loss to the service providers), the design of a near-VOD system has to meet the delay expectation of the users. For example, if users are happy to wait 5 minutes for their videos to be displayed (but would not be satisfied if they are to wait longer), then the system has to meet such delay expectation almost all the time to minimize revenue loss.

In [9], trade-off issues regarding user delay and system profitability have been addressed when the bandwidth is sufficient. Here, we consider the following bandwidth planning problem: what should be the minimum number of streams N_s^* allocated to a single video file of duration T_h in order to meet a certain user delay requirement? In this paper, we are interested in achieving a certain low probability ε that the user delay D exceeds $\hat{d}$, i.e., $P(D > \hat{d}) < \varepsilon$.

Clearly, we can always use $\lceil T_h/\hat{d} \rceil$ streams and a batching window size equal to $\hat{d}$ to satisfy the above delay requirement. However, when $\hat{d}$ is low and the arrival rate is not high, this number of streams can be excessive, and a large fraction of them may be left idle most of the time. Hence, additional saving in bandwidth can be achieved to meet the same goal.

The paper is organized as follows. In Sect. 2, we describe the batching scheme we consider in this paper (auto-gated scheduling [9]). We then address the bandwidth planning problem by first determining the appropriate window size to minimize the number of streams required, followed by determining the number of streams required. We conclude in Sect. 3.

2. AUTO-GATED SCHEDULING AND ITS BANDWIDTH PLANNING

In the auto-gated scheduling scheme [9], the first arrival following the start of the last showing of a movie begins a batching window of size W. With sufficient bandwidth, W is then the maximum delay for the users in the system. It has been shown that this batching scheme achieves good bandwidth utilization and hence would be our choice in this paper [9].

We use simulation in our study. We consider a single movie of duration T_h, with the arrival process of its requests being Poisson with rate λ. Let N_s be the number of streams allocated. All users are admitted, and since we are designing a system meeting user's delay expectation, the reneging

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

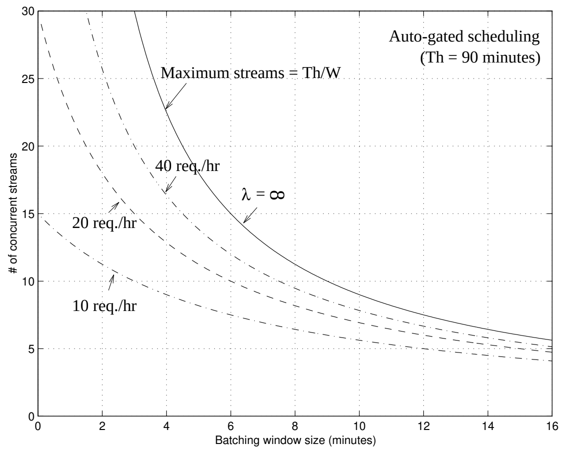

Figure 1: $\overline{S}$ versus W given λ

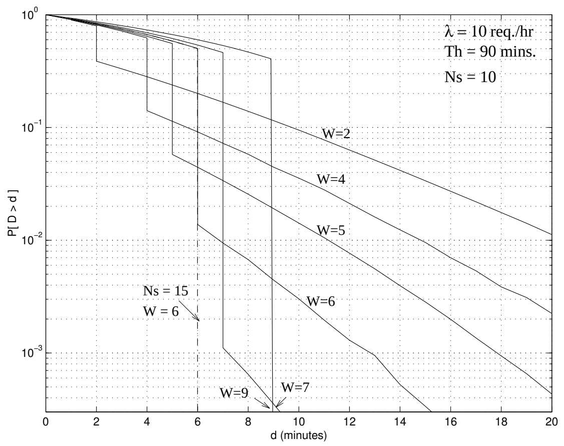

Figure 2: $P(D > d)$ with $N_s = 10$ and different W's

probability is considered to be negligible.

2.1. Sufficient bandwidth case

We first consider the case when $N_s \geq \hat{N} \triangleq \lceil T_h/W \rceil$. Since the average batch size is $1 + \lambda W$, the average number of streams used $\overline{S}$ is given by $\overline{S} = \lambda T_h/(1 + \lambda W)$[9].

Since the first user in each batch has a delay of W while the remaining ones in the batch have a delay uniformly distributed between 0 and W, the delay distribution of the users is $f_D(t) = 1/(1+\lambda W)\delta(t-W) + \lambda W/(1+\lambda W)(1/W)$, (for $0 \leq t \leq W$) where $\delta(t)$ is the impulse function with $\delta(t) = 0$ for $t \neq 0$ and $\int_{-\infty}^{\infty} \delta(t)dt = 1$. Therefore, the average user delay $\overline{D}$ is given by,

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

As W decreases, the average user delay decreases.

In Fig. 1, we plot $\overline{S}$ versus W, given λ. We consider $T_h = 90$ minutes. Also shown is the maximum T_h/W which is attained when $\lambda \to \infty$, and corresponds to the number required in the fixed scheduling case. As W increases, $\overline{S}$ decreases from λT_h, first rapidly and then slowly; most of the bandwidth reduction 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.

From the figure, we see that for low W and especially for low λ, the difference between the maximum number of streams used and $\overline{S}$ can be important, meaning that N_s can be chosen to be fewer than $\hat{N}$ without affecting much the user delay. However, such saving diminishes as W or λ increases. In this case $\overline{S}$ is almost equal to T_h/W, indicating that N_s has to be chosen close to $\hat{N}$.

2.2. Bandwidth planning

We first find the appropriate W so as to minimize the required N_s, and then find N_s^*.

As an example, let's consider meeting the delay expectation $P(D > \hat{d}) < \varepsilon$, with $\hat{d} = 6$ minutes and $\varepsilon \approx 1\%$. We show in Fig. 2 the $P(D > d)$ versus d, with $N_s = 10$ and various values of W's ($\lambda = 10$ req./hr). Also shown as the dotted line is the case for $N_s = 15$ and $W = 6$ minutes (meeting $\hat{d} = 6$ minutes). We see that for each W, there is an impulse at $d = W$ (as the first user in a batch incur a delay equal to W), followed by a long tail (due to the waiting time for an available stream). We observe that W has a big influence on user delay distribution. For $W = 4$ minutes, 1% of the users have delay higher than 15 minutes; but when W is increased to 6 minutes, 0.03% of them have delay higher than 15 minutes!

We read from the figure that, to meet our delay requirement, N_s^* is 10 (with $W = 6$ minutes) as supposed to 15 (the case of $T_h/\hat{d}$), achieving a bandwidth saving of 30%. From the figure, we see that, to minimize N_s, W should be neither larger than $\hat{d}$ (because of the impulse) nor smaller than $\hat{d}$ (because of the long tail). Therefore, we need to let

$$W = \hat{d} \tag{2}$$

in order to minimize N_s. We will hence pick such a W in the following discussion.

Note that the area under the graph is the average delay $\overline{D}$ of the users. User delay D is the sum of the gating delay D_G (the time waiting for the batching window to end) and the time waiting for an available stream D_s. As W decreases, D_G decreases while D_s increases (since streams are more likely to run out). In Fig. 3, we show $\overline{D}$ versus W, given N_s. Also shown is the average delay with infinite bandwidth according to Eq. (1). We see that indeed, as W decreases from T_h/N_s, $\overline{D}$ first decreases (due to the decrease in gating delay), and then increases (due to the waiting time for an available stream). For $N_s = 10$, we see that at minimal $\overline{D}$ (corresponding to $W = 4$ minutes), the delay distribution has an undesirable high tail (ref. Fig. 2).

We show in Fig. 4 how $P(D > \hat{d})$ varies with $\hat{d}$ given N_s. We see that as $\hat{d}$ decreases, $P(D > \hat{d})$ increases first sharply then gradually. We can also read from the graph the minimal expectation $\hat{d}$ one should target given certain N_s and ε.

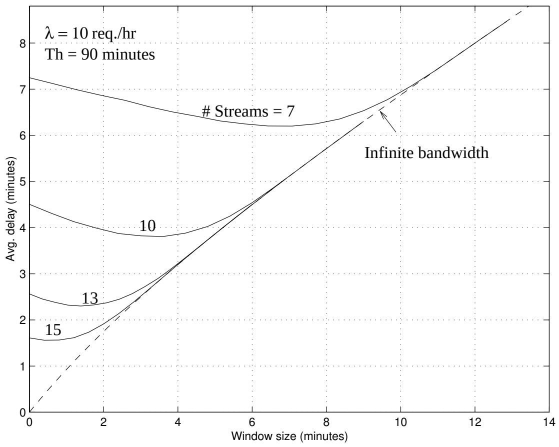

Figure 3: $\overline{D}$ versus W given N_s

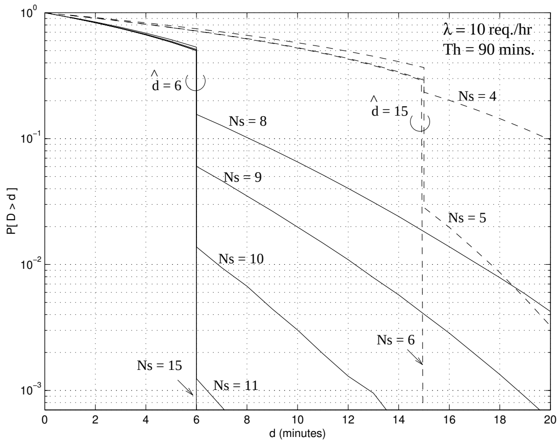

Figure 5: $P(D > d)$ versus d for different values of N_s

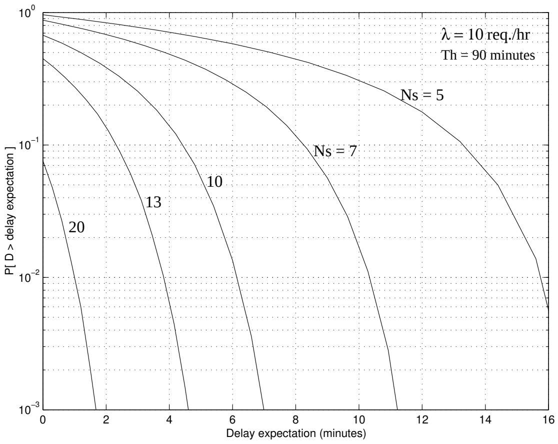

Figure 4: $P(D > \hat{d})$ versus $\hat{d}$ given N_s

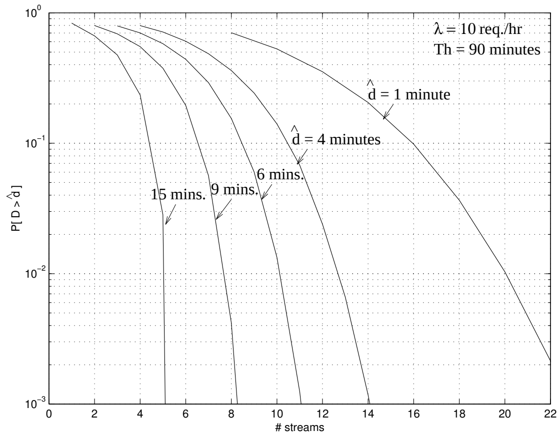

Figure 6: $P(D > \hat{d})$ versus N_s given $\hat{d}$

We show in Fig. 5 $P(D > d)$ versus d, with different values of N_s, for $\hat{d} = 6$ minutes and 15 minutes. We see that when $\hat{d}$ is equal to 15 minutes, not much bandwidth saving is gained, since the bandwidth is already well-utilized with $N_s = \hat{N} = 6$ (once again verifying that there is no much saving in bandwidth for large window size). We show in Fig. 6 $P(D > \hat{d})$ versus N_s, from which given ε and $\hat{d}$, N_s^* can be directly read.

We show in Fig. 7 N_s^* versus $\hat{d}$, given ε. Also shown is the maximum number of streams required. We see that for $\hat{d} \leq 6$ minutes, N_s^* can be markedly lower than $\hat{N}$. However, as $\hat{d}$ increases, such saving decreases quickly. When $\hat{d} \geq 12$ minutes, the saving becomes insignificant.

We show in Figs. 8 and 9 the influence of an increase in λ on $P(D > d)$ and $P(D > \hat{d})$, respectively by considering $\lambda = 30$ req./hr. Obviously, as λ increases, the delay tail increases and hence we require more streams to satisfy the same delay expectation. With the same $\hat{d}$, there is no longer as much saving as before.

We show in Fig. 10 N_s^* versus λ, when $P(D > \hat{d})$ is to be kept constant with $\hat{d} = 6$ minutes. We see that as λ decreases, N_s also decreases. As expected, as λ increases, N_s gradually increases to a value close to $\hat{N} = 15$. We see that there is not a wide range of λ in which N_s^* is much less than $\hat{N}$.

We may use the figure to find the bandwidth required for multiple movies with different request rates. For example, if there are two movies with request rates of 5 req./hr and 40 req./hr, the bandwidth required is then 22 streams as supposed to the maximum of 30 streams (given by $2\lceil T_h/\hat{d}\rceil$), achieving a bandwidth saving of more than 25%. If the bandwidth can be shared, such a saving can be even more!

3. CONCLUSIONS

Near video-on-demand achieves bandwidth saving by batching the requests for a particular video over a period of time and serving them with a single multicast stream. Since users may renege while waiting, the design of such a system has to meet user delay expectation. In this paper, we consider a single movie and specify the window size W and the minimum number of streams required N_s^* so that only a small fraction of the users are delayed beyond $\hat{d}$.

When $\hat{d}$ is low (equal to a few minutes) and the request

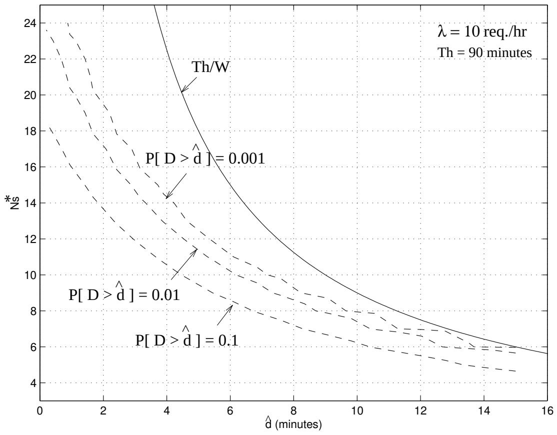

Figure 7: N_s^* versus $\hat{d}$ given ε

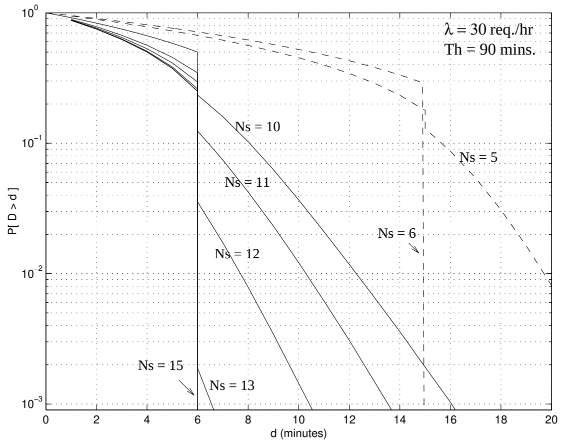

Figure 8: $P(D > d)$ versus d given N_s's for λ =30 req./hr

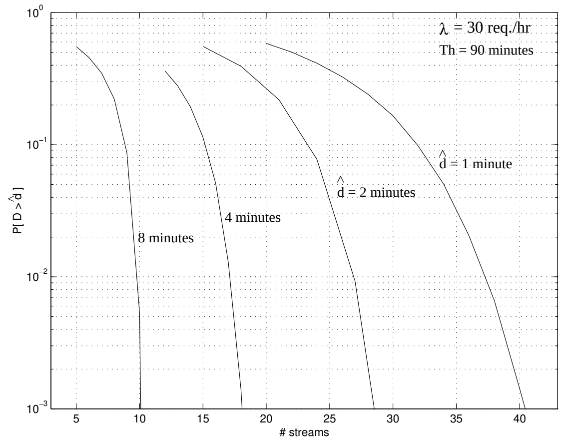

Figure 9: $P(D > \hat{d})$ versus N_s given $\hat{d}$ (λ =30 req./hr)

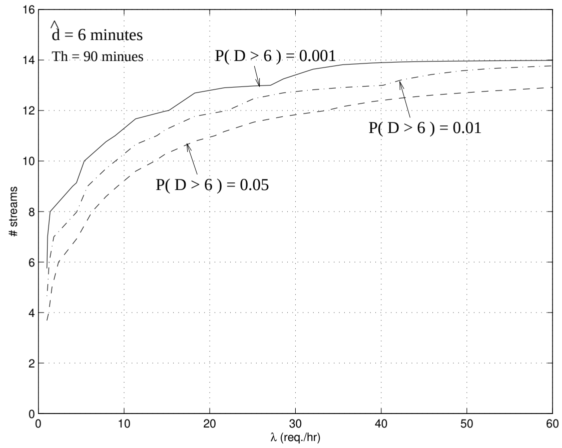

Figure 10: N_s^* versus λ given ε and $\hat{d} = 6$ minutes

rate is not high, N_s^* can be much lower than the maximum number of streams needed (equal to the ratio of the movie duration to the window size) to meet the delay expectation. We have showed that W has to be equal to $\hat{d}$ in order to minimize the stream requirement.

4. REFERENCES

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

[2] F. A. Tobagi, "Distance Learning with Digital Video," *IEEE Multimedia Magazine*, pp. 90–94, Spring 1995.

[3] K. Cleary, "Video on Demand — Competing Technologies and Services," in *Proceedings of International Broadcasting Convention*, (Amsterdam, Netherlands), pp. 432 – 437, 14–18 Sept. 1995.

[4] F. A. Tobagi, J. Pang, R. Baird, and M. Gang, "Streaming RAID^{TM} – A Disk Array Management System for Video Files," in *Proceedings of First ACM International Conference on Multimedia*, (Anaheim, CA), August 1-6 1993.

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

[6] S. Chen and M. 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), pp. 37–45, 3–6 June 1997.

[7] V. O. K. Li, W. Liao, X. Qiu, and E. W. M. Wong, "Performance Model of Interactive Video-on-Demand Systems," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1099–1109, August 1996.

[8] E. L. Abram-Profeta and K. Shin, "Scheduling Video Programs in Near Video-on-Demand Systems," in *Proceedings of ACM International Multimedia Conference*, pp. 359–371, Nov. 9-13 1997.

[9] S.-H. G. Chan, F. Tobagi, and T.-M. Ko, "Providing On-Demand Video Services Using Request Batching." to appear in *Proceedings of ICC'98*, GA, 1998.