Optimal Policies for Playing Buffered Media Streams

Steven Weber[⋆]

Drexel University, Dept. of ECE, Philadelphia, PA 19104
sweber@ece.drexel.edu

Abstract. This paper addresses a practical problem in our everyday use of streaming media on the Internet: *as a user observes the buffering of a media stream with an uncertain transfer rate, when should that user initiate playback of the stream?* The tension is that initiating playback prematurely will increase the likelihood of buffer starvation, while a delay in initiating playback is undesirable because it necessitates waiting. Three policies are studied: the optimal policy (exploiting full knowledge of the transfer process), the optimal static policy (the expected value of the optimal policy), and an online policy assuming only knowledge of the transfer rate observed thus far. Lower and upper bounds are derived on the optimal policy as well as the associated minimum cost; these bounds are expressed in terms of a (random) hitting time of the transfer process. Simulation results for a Markov modulated transfer rate process identify static and online policies as near-optimal depending on the time scale of the transfer rate process and the duration of the stream.

1 Introduction

This paper addresses a practical problem in our everyday use of streaming media on the Internet: *as a user observes the buffering of a media stream with an uncertain transfer rate, when should that user initiate playback of the stream?* The tension is that initiating playback prematurely will increase the likelihood of buffer starvation, upon which the client media player paused playback and informs the user that the client is buffering the stream. On the other hand, a delay in initiating playback is undesirable because it necessitates waiting; a key motivation behind streaming is to avoid the delay in downloading media. We will not distinguish between the cases where the playback initiation decision is made by the human user or the client media player. In either case the objective is to simultaneously minimize the prefetch time and the stall time. The prefetch time is the time between initiating the transfer and initiating playback, and the stall time is the time spent re-buffering after a buffer starvation. The difficulty in making this decision lies in the fact that *i*) the instantaneous rate of the media content is time-varying and likely unknown a priori by the client, and *ii*) the instantaneous transfer rate available on the network is time-varying, and also a

[⋆] This work is supported by the NSF under grant 0435247.

priori unknown. To simplify the problem we restrict our attention to the simpler constant bit rate media content case, leaving the uncertain transfer rate as the unknown.

Suppose the instantaneous transfer rate is given by the random process $\{x(t)\}$, and that the CBR playback rate is one. During playback, the client buffer is filling at rate $x(t)-1$ when $x(t) > 1$, and draining at rate $1-x(t)$ when $x(t) < 1$. Suppose the user incurs a cost per unit time c_p (p for prefetch) for each second between initiating transfer (at time $t = 0$) and initiating playback, and incurs a cost per unit time c_s (s for stall) for each second that playback is stalled for re-buffering. Let τ_p and τ_s be the total time spent in the prefetch and stall states respectively, so that the total cost incurred by the user is $c = c_p\tau_p + c_s\tau_s$. The client has control over τ_p by choosing when to initiate playback, but τ_s is a random variable that depends upon both τ_p and the transfer rate process. This paper investigates several policies that may be used to minimize the cost. Three policies are studied in particular:

- *Optimal policy:* using *full a priori knowledge* of the transfer process $\{x(t)\}$, the optimal policy identifies the playback initiation time that minimizes the cost c;
- *Optimal static policy:* the optimal static policy is the expected optimal playback initiation time, with the expectation taken with respect to the distribution of the transfer process;
- *Online policy:* the online policy makes a decision to initiate playback at time t based on the instantaneous transfer rate observed over $[0, t)$, with no knowledge of the future evolution of the transfer process or its distribution.

Lower and upper bounds on both the optimal playback initiation time and the corresponding minimized cost are given in terms of a (random) hitting time of the transfer process. Simulation results are provided for the three policies for the case when the transfer process is a two state Markov modulated process. The primary takeaway is that static policies are near-optimal when the timescale at which the transfer rate process changes is much smaller than the duration of the stream, while online policies are near-optimal when the timescale at which the transfer rate process changes is similar to the duration of the stream.

The rest of this paper is organized as follows. Section 2 summarizes some of the related work on streaming media. Section 3 presents the mathematical model, and Section 4 offers the analytical bounds on the optimal policy and optimal cost. Simulation results are presented in Section 5 and a brief conclusion is offered in Section 6. All proofs are placed in the appendix.

2 Related work

There are a large number of papers on the analysis, design, and performance optimization of streaming media; far too many to summarize here. Our own prior work on streaming media has focused on the network-wide optimal control of *rate adaptive* streaming media, with the objective of maximizing client-average quality of service (QoS) metrics [1, 2]. Arguably the first algorithm for media rate adaptation is the receiver-driven layered multicast (RLM) algorithm of McCanne, Jacobson, and Vetterli [3]. Selected related work on rate adaptation includes Rejaie, Handley, and Estrin [4]; Vickers, Albuquerque, and Suda [5]; Saparilla and Ross [6]; Gorinsky, Ramakrishnan, and Vin [7]; Kar, Sarkar, and Tassiulas [8]; Bain and Key [9]; Argiriou and Georgiadis [10]; and Chou and Shin [11]. Although rate adaptation introduces a valuable additional control for improving the performance of streaming media, we will not address this extension here.

There is an extensive literature starting from the late 1990s on the problem of optimal smoothing of VBR video over variable channels. In a sense this problem is orthogonal to the one we are considering, primarily because we focus on client side control whereas optimal smoothing is focused on server side control. The objective in the smoothing literature is to transmit the VBR video such that the peak to mean ratio of the transferred stream is minimized. Significant publications include Lam, Chow, and Yau [12]; Zhang, Kurose, Salehi, and Towsley [13]; Duffield, Ramakrishnan, and Reibman [14]; Rexford and Towsley [15]; and Sen, Rexford, Dey, Kurose, and Towsley [16].

There is relatively little work on client control of prefetching and playback initiation. Reisslein and Ross [17] cover prefetching policies appropriate for a server multiplexing several simultaneous streams over a shared link; Fitzek and Reisslein [18] consider a similar setup but over wireless links.

3 Mathematical model

3.1 Problem statement

A user initiates a streaming media connection at time $t = 0$. After making a connection with the media content server, the client media player begins buffering media content from the server. The control decision faced by the client is when to initiate playing the media stream, i.e., when to start draining the client buffer. It is assumed that there is a cost incurred at rate $c_p > 0$ (p for prefetch) for each second between initiating the connection and initiating playback of the content. This cost measures the user's frustration at having to wait for the content to begin playback. The transfer rate available to the client server connection is time-varying, and it is therefore possible that the client buffer will starve, forcing the client player to pause the media content. It is assumed that there is a cost incurred at rate $c_s > 0$ (s for stall) for each second the media player is forced to stall the media content after first initiating playback. Let τ_p be the time spent prefetching the stream prior to initiating playback, and let τ_s be the total time spent stalled after initiating playback. The objective of the control decision is to minimize the incurred cost $c = c_p\tau_p + c_s\tau_s$. We restrict our attention to the case where $c_p < c_s$, i.e., the unit cost of prefetching is smaller than the unit cost of stalling. This is because the optimal policy when $c_p \geq c_s$ is simply to not prefetch ($\tau_p = 0$).

3.2 Two fundamental processes

The media content is modeled as a constant bit rate (CBR) source requiring playback at unit rate, and having duration d. The instantaneous transfer rate of the client server connection is a random process $\{x(t)\}$.[1] There are two fundamental processes that determine performance: the duration of media content that *can* be played by the client and the duration of media content that *has* been played by the client.

Definition 1. The duration of media content that *can* be played by the client at time t is the cumulative amount of data transferred over $[0, t]$, up to the stream duration d:

$$y(t) = d \wedge \int_0^t x(s)\mathrm{d}s. \tag{1}$$

Definition 2. The duration of media content that *has* been played by the client is the solution of the following nonlinear differential equation

$$\dot{z}(t) = \frac{\mathrm{d}}{\mathrm{d}t} z(t) = \begin{cases} 0, & t < \tau_p \text{ or } t \geq t_z \\ 1, & y(t) > z(t) \\ x(t) \wedge 1, & \text{else} \end{cases}, \tag{2}$$

where $\tau_p = \inf\{t : z(t) > 0\}$ is the prefetch duration and $t_z = \inf\{t : z(t) = d\}$ is the time playback is completed.

The change in playback position, $\dot{z}(t)$, is zero for $t < \tau_p$ since the stream has not yet started playing, and is zero for $t \geq t_z$ since the stream has already finished. When there is unplayed media content stored at the receiver, $y(t) > z(t)$, the client is playing at the full unit rate, and $\dot{z}(t) = 1$. Otherwise $y(t) = z(t)$, which means the client buffer is starved, i.e., the client has played all currently available content. In this case the client plays the media content at rate $x(t) \wedge 1$ during starvation, hence when $x(t) < 1$ the client spends a fraction $x(t)$ of the time in play mode, and $1 - x(t)$ of the time in pause (stall) mode.

Let $t_y = \inf\{t : y(t) = d\}$ be the time that the client obtains all the media content, and let $t_z = \inf\{t : z(t) = d\}$ be the time that the client actually completes play. It is straightforward to obtain the processes $\{y(t), z(t)\}$ from the realization of $\{x(t)\}$, as illustrated in the left side of Figure 1. The play process, $\{z(t)\}$, is obtained by drawing a unit slope line starting at the playback time $t = \tau_p$ until it intersects $\{y(t)\} \wedge d$. If it hits $\{y(t)\}$ then it tracks $\{y(t)\}$ at rate $x(t) < 1$, until $x(t) > 1$ at which time it continues at unit slope, and so on. The right side of Figure 1 gives the state transition diagram of the streaming process. Here prefetch is the time after data transfer begins but before play is initiated, and postfetch is the time after data transfer finishes but before play

[1] It is easily seen that the model may be generalized to arbitrary CBR play rate r by scaling $\{x(t)\}$ by $1/r$.

is finished. It follows that the stall time, τ_s, is a functional of the play process, $\{z(t)\}$, namely:

$$\tau_s = \int_{\tau_p}^{t_z} (1 - \dot{z}(t))\mathrm{d}t. \tag{3}$$

Table 1 summarizes the mathematical notation used in the model.

Table 1. Notation

d	stream playback duration
$\{x(t)\}$	instantaneous transfer rate
$\{y(t)\}$	duration of media content that can be played
$\{z(t)\}$	duration of media content that has been played
τ_p	prefetch delay
τ_s	total time spent stalled
t_y	time the client obtains all the media content
t_z	time the client completes playback
c_p	rate cost is incurred when prefetching
c_s	rate cost is incurred when stalled

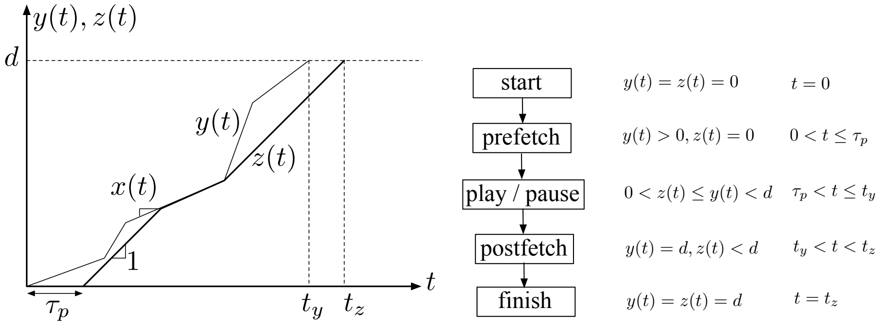

Fig. 1. Left: Evolution of the processes $\{y(t), z(t)\}$ using a prefetch duration of τ_p. Transfer of data begins at time 0 and completes at time t_y; play begins at time τ_p and completes at time t_z. **Right:** State transitions in the streaming process, along with the times at which they occur and the values of $(y(t), z(t))$ that characterize each state.

3.3 Three prefetching policies

Optimal policy: under the assumption that $c_p < c_s$, it follows that the optimal policy to minimize $c = c_p\tau_p + c_s\tau_s$ is to set $\tau_p^{op} = \inf\{t : \tau_s = 0\}$ (op for optimal policy). The optimal policy clearly requires a priori knowledge of the transfer rate process $\{x(t)\}$.

Optimal static policy: the optimal static policies is the expected value of the optimal policy, $\tau_p^{sp} = \mathbb{E}[\tau_p^{op}]$ (sp for static policy).
Online policy: let $\{\mathcal{F}_t\}$ be the filtration of knowledge available to the client by time t. The online policy is to select a prefetch time $\tau_p^{ol} = t$ based on $\{\mathcal{F}_t\}$ (ol for online).

4 Analytical results

4.1 Optimal policy

Consider the case where the transfer rate process $\{x(t)\}$ is known a priori by the client, as assumed by the optimal policy. We begin with an easy result for the case when $x(t) = x$ for all t. All proofs are in the appendix.

Lemma 1. Suppose $x(t) = x$ for all t. The optimal policy and the corresponding optimal cost are

$$\tau_p^{op} = d\frac{(1-x)^+}{x}\mathbf{1}_{c_p<c_s}, \quad c^{op} = (c_p \wedge c_s)d\frac{(1-x)^+}{x}. \tag{4}$$

Next consider the case when either $x(t) < 1$ for all t or $x(t) > 1$ for all t.

Lemma 2. Suppose $x(t) < 1$ for all t or $x(t) \geq 1$ for all t. The optimal policy and the corresponding optimal cost are

$$\tau_p^{op} = (t_y - d)^+\mathbf{1}_{c_p<c_s}, \ c^{op} = (c_p \wedge c_s)(t_y - d)^+, \tag{5}$$

where $t_y = \inf\{t : y(t) = d\}$ is the transfer completion time.

Note that $x(t) \geq 1$ for all t implies $t_y \leq d$, while $x(t) < 1$ for all t implies $t_y > d$. In the above regimes the optimal policy and corresponding cost are expressible in terms of the single random variable t_y. The point to emphasize is that the t_y is a hitting time of the transfer process $\{y(t)\}$, and is independent of the play process $\{z(t)\}$.

Unfortunately the simple characterization of Lemma 2 does not hold for general $\{x(t)\}$, but we can obtain bounds on the cost in terms of t_y. In the following theorem we restrict our attention to instantaneous transfer rate processes with minimum and maximum rates, $x_{\min}$ and $x_{\max}$, respectively. Without loss of generality we assume $x_{\min} < 1 < x_{\max}$, since otherwise the process falls under the class of processes addressed in Lemma 2.

Theorem 1. Suppose $\{x(t)\}$ is bounded from below and above such that $x_{\min} < x(t) < x_{\max}$, where $x_{\min} < 1 < x_{\max}$. Define the maximum prefetch time under the lower and upper bounds as

$$\tau_p^{\mathrm{lmax}} = (t_y - d)^+, \quad \tau_p^{\mathrm{umax}} = \frac{1 - x_{\min}}{x_{\max} - x_{\min}}(x_{\max}t_y - d). \tag{6}$$

The optimal policy and the corresponding optimal cost are

$$\tau_p^{op} \in [\tau_p^{\mathrm{lmax}}\mathbf{1}_{c_p<c_s}, \tau_p^{\mathrm{umax}}\mathbf{1}_{c_p<c_s}], \tag{7}$$
$$c^{op} \in [c_p \wedge c_s\tau_p^{\mathrm{lmax}}, c_p \wedge c_s\tau_p^{\mathrm{umax}}]. \tag{8}$$

Note that the bounds become tight as $x_{\max} \downarrow 1$.

4.2 Optimal static policy

This policy is appropriate for the regime when the transfer rate process $\{x(t)\}$ is unknown a priori by the client, but the statistics of $\{x(t)\}$ are known (or may be estimated through observation of multiple streaming sessions), permitting the computation of the optimal static prefetch policy: $\tau_p^{sp} = \mathbb{E}[\tau_p^{op}]$. The optimal static policy may be approximated by observing n iid realizations of the optimal prefetch policy, $\tau_p^{op,1}, \ldots, \tau_p^{op,n}$ for n realizations of the transfer process, $\mathbf{x}_1, \ldots, \mathbf{x}_n$, (where $\mathbf{x}_i = \{x_i(t)\}$), and computing $\hat{\tau}_p^{sp} = \frac{1}{n} \sum_{i=1}^n \tau_p^{op,i}$.

4.3 Online policy

We now assume the transfer rate process $\{x(t)\}$ is unknown by the client a priori. Lacking any information as to the distribution of the process $\{x(t)\}$, we seek to estimate the data transfer completion time, t_y, at each time t, using only knowledge of $\{x(u)\}_0^t$. An estimate of a quantity w is denoted as $\hat{w}$, and the notation $\hat{w}(t)$ indicates that this is the estimate made using knowledge of $\{x(u)\}_0^t$. The empirical estimate of the average transfer rate is:

$$\hat{x}(t) = \frac{1}{t} \int_0^t x(u) \mathrm{d}u = \frac{y(t)}{t}. \tag{9}$$

The estimate of the transfer completion time as of time t is $\hat{t}_y(t) = t \frac{d}{y(t)}$. Define $\hat{\tau}_p^{op}(t)$ as the best estimate as of time t of the time to start playback so as to avoid stalling. It is clear that

$$\hat{\tau}_p^{op}(t) = \hat{t}_y(t) - d = d\Big(\frac{1}{\hat{x}(t)} - 1\Big). \tag{10}$$

The online policy is then:

$$\tau_p^{ol} = \inf\{t : \hat{\tau}_p^{op}(t) < t\}, \tag{11}$$

i.e., to begin playback at the first time that the current time exceeds the estimated optimal prefetch time.

5 Simulation results

To investigate the performance of the three policies we consider the case when the transfer process $\{x(t)\}$ is modulated by a homogeneous continuous time two state Markov process. In particular, let the states be 1 and 2, where $x(t) = r$ when the process is in state 1 and $x(t) = 2 - r$ when the process is in state 2 for some $r \in [0, 1]$. Let the rate matrix for the Markov chain be

$$Q = \begin{bmatrix} -q & q \\ q & -q \end{bmatrix}, \tag{12}$$

so that the stationary distribution is $\pi_1 = \pi_2 = \frac{1}{2}$, and q governs the timescale of the evolution of the chain. Note that the average transfer rate is $\mathbb{E}[x(t)] = \pi_1 r + \pi_2(2-r) = 1$. We explore the performance of the three policies in the regime when $q \ll d$ (the transfer process evolves on a slower timescale than the stream duration), as well as $q \gg d$ (the transfer process evolves on a faster timescale than the stream duration). Further we explore the performance of the three policies in the regime when $r \approx 0$ (the transfer process is bursty with rates ≈ 0 and ≈ 2) and $r \approx 1$ (the transfer process is smooth with rates ≈ 1). Because of this we refer to q as the *temporal burstiness parameter* and r as the *spatial burstiness parameter*.

Figure 2 presents the simulation results. All four plots show the average cost $\bar{c} = c_p \bar{\tau}_p + c_s \bar{\tau}_s$ under the three policies (optimal, static, online), for

$$d = 10 \; c_p = 1 \; c_s = 2. \tag{13}$$

For r small the transfer rate process is spatially bursty while for r near one the transfer rate process is almost (spatially) constant. For q small the transfer rate process is temporally almost static, while for q large the transfer rate process is temporally bursty. The top two plots have the transfer rate r as the abscissa, for transition rate $q = .1$ (top) and $q = 10$ (second). The top two plots demonstrate that online policy is superior for temporally static transfer rates while the static policy is superior for temporally bursty processes. Intuitively, knowledge of the statistics of the process are only of value when there are sufficient number of rate changes so that the fluctuations will average out. The bottom two plots have the transition rate q as the abscissa, for transfer rate $r = 0$ (third) and $r = 0.9$ (bottom). The bottom two plots demonstrate that fact that there is a critical transition rate q_c such that for $q < q_c$ the online policy is superior, while for $q > q_c$ the static policy is superior. Furthermore, q_c is larger for spatially bursty processes (r small) and smaller for spatially smooth processes (r large). All four plots demonstrate that the average cost is monotone decreasing as process becomes spatially and temporally smoother, i.e., as q and r increase.

Finally, Figure 3 presents the optimal (no stall) prefetch time along with the lower and upper bounds from Theorem 1 versus the transfer rate parameter r. The bounds are seen to be reasonably tight for all r, and are seen to become tight as $r \uparrow 1$.

6 Conclusion

In this paper we have studied policies governing when a streaming media client should commence playback. The tradeoff that is optimized is between minimizing the stall time (due to buffer starvation) and minimizing the prefetch time. Three policies were considered: the optimal policy (assuming full knowledge), the optimal static policy, and an online policy. It was observed that the online policy is superior when the duration of the stream is on the order of the fluctuations of the transfer process, while the static policy is superior when the duration of the stream is much longer than the fluctuation timescale.

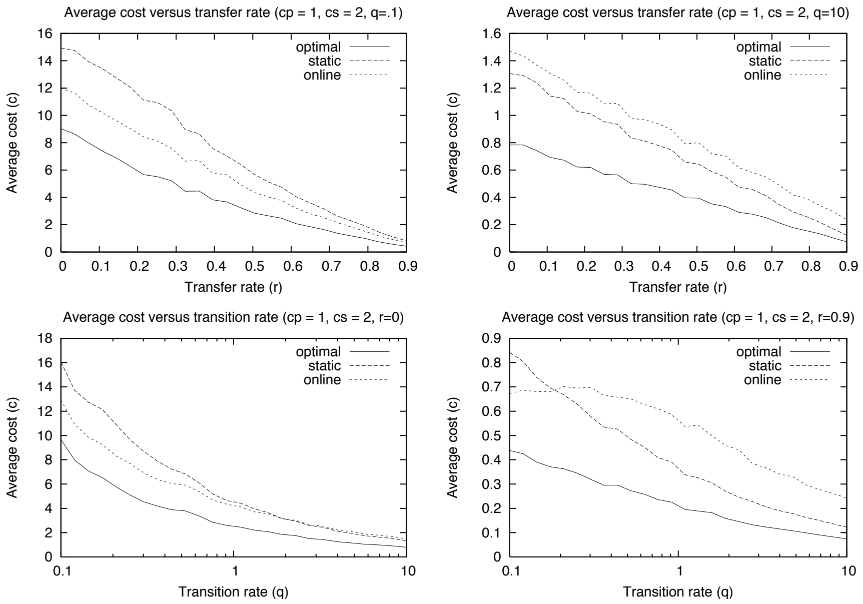

Fig. 2. Average cost $\bar{c} = c_p\bar{\tau}_p + c_s\bar{\tau}_s$ under the three policies (optimal, static, online). The top two plots have the transfer rate r as the abscissa, for transition rate $q = .1$ (top) and $q = 10$ (second). The bottom two plots have the transition rate q as the abscissa, for transfer rate $r = 0$ (third) and $r = 0.9$ (bottom).

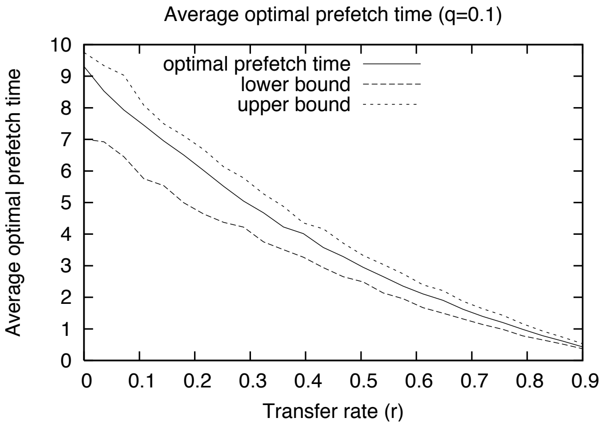

Fig. 3. The optimal (no stall) prefetch time along with the lower and upper bounds from Theorem 1 versus the transfer rate parameter r.

References

1. Weber, S., de Veciana, G.: Rate adaptive multimedia streams: optimization and admisssion control. IEEE/ACM Transactions on Networking **13**(6) (December 2005) 1275–1288
2. Weber, S., de Veciana, G.: Flow-level QoS for a dynamic load of rate adaptive sessions sharing a bottleneck link. accepted for publication in Computer Networks (October 2006)
3. McCanne, S., Jacobson, V., Vetterli, M.: Receiver-driven layered multicast. In: Proceedings of ACM SIGCOMM, Stanford, CA (August 1996)
4. Rejaie, R., Handley, M., Estrin, D.: Quality adaptation for congestion controlled video playback over the internet. In: Proceedings of ACM SIGCOMM. (1999) 189–200
5. Vickers, B., Albuquerque, C., Suda, T.: Source-adaptive multi-layered multicast algorithms for real-time video distribution. Technical Report Technical Report ICS-TR 99-45, University of California, Irvine (June 1999)
6. Saparilla, D., Ross, K.: Optimal streaming of layered video. In: Proceedings of IEEE INFOCOM. (March 2000)
7. Gorinsky, S., Ramakrishnan, K., Vin, H.: Addressing heterogeneity and scalability in layered multicast congestion control. Technical Report TR2000–31, UT-Austin (November 2000)
8. Kar, K., Sarkar, S., Tassiulas, L.: Optimization based rate control for multirate multicast sessions. In: Proceedings of IEEE INFOCOM. (2001) 123–132
9. Bain, A., Key, P.: Modeling the performance of in-call probing for multi-level adaptive applications. Technical Report Technical Report MSR-TR-2002-06, Microsoft Research (October 2001)
10. Argiriou, N., Georgiadis, L.: Channel sharing by rate-adaptive streaming applications. In: Proceedings of IEEE INFOCOM. (2002)
11. Chou, C.T., Shin, K.G.: Analysis of adaptive bandwidth allocation in wireless networks with multilevel degradable quality of service. IEEE Transactions on Mobile Computing **3**(1) (January–February 2004) 5–17
12. Lam, S., Chow, S., Yau, D.: A lossless smoothing algorithm for compressed video. IEEE/ACM Transactions on Networking **4**(5) (October 1996) 697–708
13. Zhang, Z.L., Kurose, J., Salehi, J.D., Towsley, D.: Smoothing, statistical multiplexing, and call admission control for stored video. IEEE Journal on Selected Areas in Communications **15**(6) (August 1997) 1148–1166
14. Duffield, N., Ramakrishnan, K., Reibman, A.: SAVE: an algorithm for smoothed adaptive video over explicit rate networks. IEEE/ACM Transactions on Networking **6**(6) (December 1998) 717–728
15. Rexford, J., Towsley, D.: Smoothing variable-bit-rate video in an internetwork. IEEE/ACM Transactions on Networking **7**(2) (April 1999) 202–215
16. Sen, S., Rexford, J., Dey, J., Kurose, J., Towsley, D.: Online smoothing of variable-bit-rate streaming video. IEEE Transactions on Multimedia **2**(1) (March 2000) 37–48
17. Reisslein, M., Ross, K.W.: High-performance prefetching protocols for VBR prerecorded video. IEEE Network (November/December 1998) 46–55
18. Fitzek, F., Reisslein, M.: A prefetching protocol for continuous media streaming in wireless environments. IEEE Journal on Selected Areas in Communications **19**(10) (October 2001) 2015–2028

Proof of Lemma 1. It is obvious that not prefetching is optimal when $x \geq 1$. Suppose instead that $x < 1$. It is clearly sub-optimal to have any postfetch time, thus the maximum prefetch time is $\tau_p^{\max} = d\frac{1-x}{x}$. Under prefetch time $\tau_p \in [0, \tau_p^{\max}]$ the stream plays at rate 1 for $\tau_p \frac{x}{1-x}$ seconds, and plays at rate x for $\frac{d}{x} - \frac{\tau_p}{1-x}$ seconds. The time spent playing at rate x can be decomposed into $d - \tau_p \frac{x}{1-x}$ seconds spent playing at rate 1 and $d\frac{1-x}{x} - \tau_p$ seconds spent stalled. The cost incurred is then

$$c = c_p \tau_p + c_s \left(d\frac{1-x}{x} - \tau_p \right) = (c_p - c_s)\tau_p + c_s d\frac{1-x}{x}.$$

The cost is linear in τ_p with slope $c_p - c_s$ and intercept $c_s d\frac{1-x}{x}$. When $c_p < c_s$ the cost is minimized by setting $\tau_p^* = \tau_p^{\max}$, incurring a cost of $c^{\min} = c_p d\frac{1-x}{x}$. ■

Proof of Lemma 2. It is obvious that not prefetching is optimal when $x(t) \geq 1$ for all t. Suppose instead that $x(t) < 1$ for all t. Since $x(t) < 1$ for all t it follows that there exists some $t^* \in [\tau_p, t_y]$ marking the first time the buffer starves. In particular, $y(t) > z(t)$ for all $t \in [0, t^*]$ and $y(t) = z(t)$ for all $t \in [t^*, t_y]$. The buffer stays starved due to the assumption that $x(t) < 1$ for all t. The client plays at full unit rate throughout $[\tau_p, t^*]$, and plays at rate $x(t)$ throughout $[t^*, t_y]$. As the total time spent playing must sum to d it follows that $d - (t^* - \tau_p)$ seconds are spent playing in $[t^*, t_y]$, while the remaining $(t_y - t^*) - (d - (t^* - \tau_p)) = t_y - d - \tau_p$ seconds are spent stalled. It is clear that for a given t_y and d there will be an unnecessary postfetch period if $\tau_p > t_y - d$, thus we can without loss of generality restrict our attention to prefetch durations in the interval $\tau_p \in [0, t_y - d]$. For any such prefetch duration the incurred cost is

$$c = c_p \tau_p + c_s(t_y - d - \tau_p) = (c_p - c_s)\tau_p + c_s(t_y - d).$$

From here the optimal policy is easily identified using an argument identical to that in the proof of Lemma 1. ■

Proof of Theorem 1. The lower bound on the stall time occurs when the data arrives as early as possible. Similarly, the realization of $\{y(t)\}$ that maximizes the stall time is for the data to arrive as late as possible. A little thought shows that the maximum useful prefetch time for the lower bound arrival pattern is $\tau_p^{\mathrm{lmax}} = t_y - d$. Assume $\tau_p \in [0, \tau_p^{\mathrm{lmax}}]$. The best realization of $\{y(t)\}$ for a given file transfer completion time, t_y, is for the data to arrive at the maximum rate, $x_{\max}$, for as long as possible, then at the minimum rate, $x_{\min}$, for the remaining time. The playback process $\{z(t)\}$ under prefetch time τ_p can then be computed for this best case. It is clear that there will not be a postfetch period under this realization, and that $\{z(t)\}$ plays at full unit rate for the entire duration except for the last τ seconds, where the value of τ can be computed as the solution of $t_y - \tau_p - \tau = d - x_{\min}\tau$, yielding $\tau = \frac{t_y - \tau_p - d}{1-r}$. Thus the minimum time spent in the stall state is $\tau_s^{\min} = \tau(1-r) = t_y - \tau_p - d$.

Next consider the worst case realization of $\{y(t)\}$ shown in the left figure in Figure 4. The first step is to identify the maximum useful prefetch time. See the right figure in Figure 4. The maximum useful prefetch time is the shortest time such that it is not possible to stall playback, given the data transfer completion time t_y. The worst case realization of $\{y(t)\}$ subject to the assumed bounds is shown in the figure. The maximum useful prefetch time is then the smallest τ_p such that beginning $\{z(t)\}$ at τ_p never touches $\{y(t)\}$. From the figure it can be seen that this maximum time is $\tau_p^{\text{umax}} = \frac{1-x_{\min}}{x_{\max}-x_{\min}}(x_{\max}t_y - d)$. The worst realization is for the data to arrive at the minimum rate, $x_{\min}$, for as long as possible, then at the maximum rate, $x_{\max}$, for the remaining duration. Tedious algebra yields the time where $\{z(t)\}$ plays at rate $x_{\min}$ is $\frac{x_{\max}t_y-d}{x_{\max}-x_{\min}} - \frac{\tau_p}{1-x_{\min}}$, and thus the maximum time spent in the stall state is $\tau_s^{\max} = \tau_p^{\text{umax}} - \tau_p$.

Combining the bounds on the stall time leads to bounds on the cost:

$$\begin{aligned} c^{\text{lmin}} &= c_p\tau_p + c_s\tau_s^{\min} \\ &= (c_p - c_s)\tau_p + c_s\tau_p^{\text{lmax}}, \quad \tau_p \in [0, \tau_p^{\text{lmax}}] \\ c^{\text{umin}} &= c_p\tau_p + c_s\tau_s^{\max} \\ &= (c_p - c_s)\tau_p + c_s\tau_p^{\text{umax}}, \quad \tau_p \in [0, \tau_p^{\text{umax}}] \end{aligned}$$

Under both bounds the optimal policy is again obvious, namely to not prefetch if $c_p > c_s$, and to prefetch for $\tau_p = \tau_p^{\max}$ if $c_p < c_s$. ■

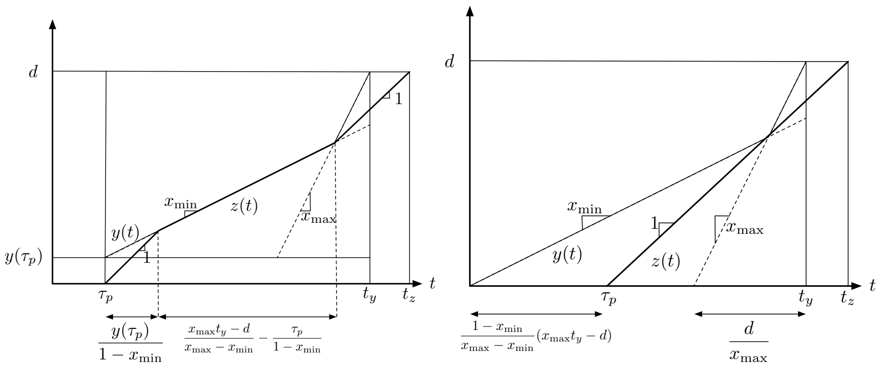
d
$y(\tau_p)$
$x_{\min}$
$z(t)$
$x_{\max}$
$y(t)$
1
1
τ_p
t_y
t_z
t
$\frac{y(\tau_p)}{1-x_{\min}}$
$\frac{x_{\max}t_y - d}{x_{\max}-x_{\min}} - \frac{\tau_p}{1-x_{\min}}$
d
$x_{\min}$
1
$x_{\max}$
$y(t)$
$z(t)$
τ_p
t_y
t_z
t
$\frac{1-x_{\min}}{x_{\max}-x_{\min}}(x_{\max}t_y - d)$
$\frac{d}{x_{\max}}$

Fig. 4. Sketch of proof of Theorem 1. **Left:** upper bound on the stall time. **Right:** maximum useful prefetch time for upper bound.