Optimal Streaming of Layered Video

Despina Saparilla Keith W. Ross

Abstract— **This paper presents a model and theory for streaming layered video. We model the bandwidth available to the streaming application as a stochastic process whose statistical characteristics are unknown a priori. The random bandwidth models short term variations due to congestion control (such as TCP-friendly conformance). We suppose that the video has been encoded into a base and an enhancement layer, and that to decode the enhancement layer the base layer has to be available to the client. We make the natural assumption that the client has abundant local storage and attempts to prefetch as much of the video as possible during playback. At any instant of time, starvation can occur at the client in either of the two layers. We study the dynamic allocation of the available bandwidth to the two layers. For the case of an infinitely-long video, we find that the optimal policy takes on a surprisingly simple and static form. For finite-length videos, the optimal policy is a simple static policy when the enhancement layer is deemed at least as important as the base layer. When the base layer is more important, we design a threshold policy heuristic which switches between two static policies. We provide numerical results that compare the performance of no-prefetching, static and threshold policies.**

I. Introduction

In recent years, streaming stored video has become a popular Internet application [1–4]. We expect the traffic emerging from streaming stored video to be a major, if not dominant, Internet traffic type in the upcoming years because (i) like the Web, it is an intrinsically appealing application, (ii) each video stream generates a relatively large amount of traffic, and (iii) increased deployment of high-speed residential access networks (e.g., cable modems and ADSL) will permit a greater number of users to stream video at high rates.

One major technological trend that should be taken into account in the design of streaming stored video applications is the phenomenal increase of disk capacity at local client machines. Today, standard PCs are being sold with tens of gigabytes, and if the current growth trend continues they may be sold with hundreds of gigabytes in upcoming years. This immense local storage capacity fully opens the door to prefetching video during client playback.

The Internet itself also has three characteristics that need to be taken into account when designing video streaming applications. First, the Internet provides its users with highly heterogeneous access rates. Second, the traffic load over a link can wildly fluctuate over a broad range of time scales [5]. And third, currently the dominant traffic type is TCP, which has been designed to share bandwidth with other traffic flows by appropriately limiting its transmission rate. The first two characteristics strongly suggest the use of an adaptive transmission scheme at the server, such as transmission of layered-encoded video. The third characteristic suggests that streaming video should be designed to cooperate fairly with existing TCP flows.

In this paper we develop a model that provides a framework for high-level design of streaming stored video applications. We develop the model in the current context of abundant local storage, heterogeneous user access rates, fluctuating traffic load on links, and the need for the application to conform to a congestion control mechanism (such as TCP-friendly conformance). Given that there is abundant local storage, we naturally allow for limitless prefetching during client playback. Our theory permits the video to be VBR-encoded, although the results remain insightful for the special case of CBR video. The model supposes that the bandwidth available to the video streaming application is variable; it could, for example, be the fair-share bandwidth determined by a TCP-friendly algorithm [6–8].

We also suppose that the video is layered encoded. Layered encoding is useful in order to cope with the heterogeneity of user access rates and with the competing traffic in the links between server and client. In this paper we suppose that the video is encoded in two layers – a base layer and an enhancement layer. At any instant of time, starvation can occur at the client in either of the two layers. During periods of starvation, the client applies video error concealment to hide the loss [9]. The fundamental problem that we address in this paper is the dynamic allocation of the available bandwidth to the two layers in order to minimize the impact of client starvation. A conservative policy allocates all the available bandwidth to the base layer until the entire base layer has been prefetched (at which the available bandwidth is allocated to the enhancement layer); a more aggressive, optimistic policy is to allocate the available bandwidth in proportion to the average consumption rates of the layers. The problem of dynamically allocating bandwidth among the layers can be formulated as an adaptive stochastic control problem [10]. The fraction of bandwidth allocated to a layer can

depend on a number of observable factors, including the current and past available bandwidth, the current prefetch buffer contents, and the dynamic consumption rates of the videos. However, the statistical characteristics of the available bandwidth (e.g., mean and variance) are not given a priori to the client-server system.

We study this dynamic allocation problem for two cases: the case of an infinite-length video, which approximates the important case of a long video with limited or no user repositioning (as would be the case in a movie); and the finite video case, which models the case of a shorter video clip. For the infinite video case we find that the optimal policy is surprisingly simple. It is a static policy that allocates a constant fraction of the bandwidth to each layer throughout the transmission of the video. Although making extensive use of prefetching, static policies do not need to take into account current prefetch buffer contents. For the finite video case, we find that the nature of the optimal policy depends on the relative importance of the various layers. When the enhancement layer is deemed as important as the base layer, then the optimal policy is shown to be a specific static policy. However, when the base layer is relatively more important, then static policies are suboptimal and, in fact, can perform poorly. For this important case, we devise a simple heuristic which switches between two static policies when the base-layer prefetch buffer exceeds a threshold. We provide numerical results which show that threshold policies can provide significantly better performance than static policies. The numerical results also illustrate the importance of prefetching.

The paper is organized as follows. In Section 2 we provide further motivation for streaming layered-video and prefetching. In Section 3 we precisely define the model. In Section 4 we define and solve the problem of optimally allocating available bandwidth to the base and enhancement layers for infinitely-long video. In Section 5 we study a similar problem for finite-length video. We develop heuristics for the finite-length case and provide simulation results in Section 6.

II. Streaming Stored Video

One fundamental property of stored video, as mentioned in the Introduction and observed in many other papers [11–17], is that it is prefetchable. Prefetching is advantageous for at least three reasons. First, it allows the client to locally build up a reservoir in preparation for future bandwidth droughts. Droughts can occur over short time scales due to bursty Web requests, congestion avoidance in competing TCPs, and the variable-bit rate transmissions in competing video streams. Bandwidth droughts can also occur on longer time scales due to changes in the number of competing streams and Web surfers, and due to route changes. A second motivation for prefetching is that when the video stream is variable-bit-rate (VBR) encoded, then future high-bit rate scenes can be prefetched when there is excess available bandwidth. Finally, a third motivation is to reduce (or eliminate) the re-buffering delay when the user repositions playback at a point into the future.

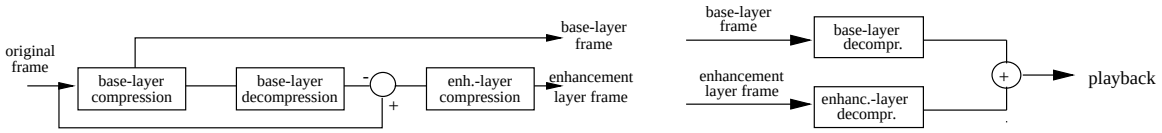
A second property of video is that it is loss (i.e., starvation) tolerant. Sender-side (e.g., FEC) and receiver-side (e.g., block repetition, prediction, interpolation) [9] techniques can be used to reduce the visual effects of loss. A third property of video is that it is often VBR encoded. This implies that the rate at which the video data is drained at the client fluctuates over many different time scales. However, because the video is prerecorded and stored, the rate fluctuations are known *a priori* to the server.

When designing an application for streaming stored video, we must also take into account the nature of the Internet. Access rates to the Internet vary by several orders of magnitude. Many users are restricted to dial-up modem rates of 56 Kbps or less, whereas other users have 100 Mbps Ethernet access. Furthermore, the competing network traffic load between server and client can widely fluctuate over many different time scales.

These two Internet characteristics – heterogeneous access rates and fluctuating network traffic – motivate the use of layered encoding. With two layers, it may be possible to quickly prefetch the base layer so that it is immediately available after user repositioning. With layered encoded video, when the long-term average available bandwidth is insufficient to support all the layers, the server does not transmit higher layers, which results in lower but often acceptable quality for the user. *A critical property of layered encoding is that in order to decode a layer, all the lower layers must also be present at the client.*

Fig. 1 illustrates how video is typically encoded into two layers. First the video is compressed into a base layer. Next, the base layer is de-compressed and subtracted from the original uncompressed video. This difference is then compressed to form the enhancement layer. At the receiver, the layers are independently de-compressed and then added together. If packet loss occurs for either layer, the client can attempt to conceal the loss using, for example, block repetition, prediction and interpolation.

Another important characteristic of today's Internet is that dominant traffic types (HTTP, SMTP, NNTP, etc.) run over TCP. TCP uses a congestion control mechanism that forces connections to exhibit fair behavior [18]. Streaming video applications should be designed to be cooperative with the TCP connections by reacting to congestion [19]. This can be done, for example, by probing to dis-



(a) Sender (b) Receiver

Fig. 1. Layered encoding and decoding of video.

cover the fair-share of network bandwidth and transmit at a rate that does not exceed this fair share. Applications with this property are said to be "TCP friendly" [6, 7]. An application's fair share rate can be estimated by its round-trip times and its loss rates [6–8].

A. Related Research

Rejaie et al [20, 21] consider a broad range of architectural issues for streaming layered encoded video. They argue for the need for end-to-end congestion control, quality adaptation and error control for streaming applications. Their analysis assumes that (1) the congestion control mechanism employs an additive increase multiplicative decrease (AIMD) algorithm, (2) the video is encoded in many layers, (3) the encoding is CBR. Furthermore, they do not account for error concealment at the receiver, so a complete layer must be available at the receiver to make use of it. In the context of these assumptions, they develop buffer allocation mechanisms that meet natural QoS goals. Although our paper is similar in spirit to [21], the model and the approach differs in many respects. Our model allows for (1) a general evolution of the available bandwidth (rather than one based on the AIMD algorithm), (2) partial loss and error concealment at the receiver, and (3) VBR as well as CBR encoded video. Our approach also differs in that we formulate the problem as an optimal stochastic control problem, and study the problem for both long and short videos. Our goal is to gain fundamental insight into the streaming of layered video in a broad context.

Podolsky et al [22] also formulate an interesting optimization problem for streaming layered video. In their model, the bandwidth between server and client is constant, but packets are independently lost with a constant probability. They do not explicitly consider extensive client prefetching nor TCP-compliant transmission schedules. Their focus is on optimal retransmission of lost packets from the different layers.

III. The Model

Video is stored in a server and is to be streamed across the Internet to a client. Let the length (in seconds) of the video be denoted by T. We suppose that the video is VBR layered-encoded into a base layer and an enhancement layer. Although we allow for VBR encoding of each layer, the theory developed here remains insightful for the case of CBR-encoded video. To simplify the notation, we use a fluid model to represent the streaming of the encoded video. (This theory can be converted to its discrete equivalent without significant modification.) Let $r_b(t)$ denote the encoded rate of the base layer t seconds into the video; similarly define $r_e(t)$ for the enhancement layer.

Without loss of generality, we suppose that the client begins to playback the video at time $t = 0$. Initially, we exclude interactive actions such as pause/resume and repositioning. Thus at time t the client desires to consume base layer video at rate $r_b(t)$ and enhancement layer video at rate $r_e(t)$. To remove jitter and short time scale bandwidth variations, most streaming systems build up a few seconds of video before playback [1, 2]. Our model also allows for an initial playback delay, denoted by Δ. Since playback begins at time $t = 0$, a playback delay of Δ seconds means that the client requests the video at time $t = -\Delta$. Throughout this paper, we suppose that Δ is a fixed parameter (e.g., four seconds). We make the approximation that the delay between the server and the client is zero; this is a reasonable approximation since RTTs are relatively small.

Let $X(t)$, $-\Delta \le t$, be the rate available to the stream at time t. The server might determine the rate $X(t)$, for example, from RTTs and packet loss rates using one of the TCP-friendly procedures [6–8]. The available bandwidth $X(t)$ can vary on short time scales due to competing Web transfers, competing VBR streams, and competing TCPs using congestion avoidance; it can also vary on long time scales due to changes in the number of streams and users, and due to route changes. At time t the server knows the current available bandwidth $X(t)$ and its past values, but has no knowledge of its future values (although it can try to predict them from the current and past values). We view $\{X(t), t \ge -\Delta\}$ as a stochastic process.

We suppose that the server always transmits at the rate allowed by the available bandwidth. When the available bandwidth exceeds the aggregate consumption rate, the system is prefetching into the client storage, which we model as infinite. We also suppose that the server never

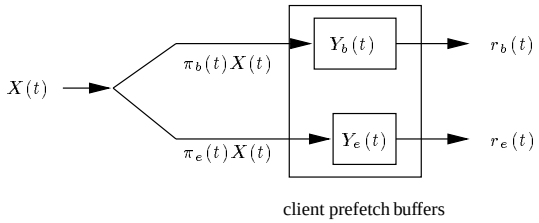


Fig. 2. Allocating the available bandwidth to the two layers.

transmits data that have already missed their deadline for timely consumption. Thus at time t the server transmits video at rate $X(t)$, and all of the transmitted video will eventually be consumed by the client.

At each time instant t the server must allocate the available bandwidth $X(t)$ among the base and enhancement layers. Let $\pi_b(t)$ and $\pi_e(t)$ denote the fraction of $X(t)$ that the server allocates to the base and enhancement layers, respectively. Of course, $\pi_b(t) + \pi_e(t) = 1$ for all t. We refer to $\pi = (\pi_b(t), t \geq -\Delta)$ as the *streaming policy*. As shown in Fig. 2, at time t the base-layer prefetch buffer in the client is fed at rate $\pi_b(t)X(t)$ and, when nonempty, is drained at rate $r_b(t)$. An analogous statement is true for the enhancement layer. Note that the client prefetch buffers comprise a system of two fluid queues whose occupancy depends on $X(t)$ and the prefetch policy π.

Throughout this paper we suppose that the server is aware of the amount of data in the prefetch buffers. In practice, the server could accurately estimate the buffer contents from receiver reports. For example, if the the server receives a report stating that at time t the contents are $Y_b(t)$ and $Y_e(t)$, then it can estimate the contents at time $t + \delta$ as

$$Y_b(t+\delta) \approx Y_b(t) + \int_{s=t}^{t+\delta} [\pi_b(s)X(s) - r_b(s)]ds.$$

We consider prefetch policies in a general sense. The policy allocation $\pi_b(t)$ can depend on t, on $X(t)$ and its entire past history $X(s), s < t$, and on the past policy allocations $\pi_b(s), s < t$. Because $Y_b(t)$ and $Y_e(t)$ are uniquely defined by $X(s), s \leq t$ and $\pi_b(s), s \leq t$, the policy can depend on the current and past prefetch buffer contents as well. However, we make the natural assumption that there is no a priori statistical characterization of $\{X(t), t \geq -\Delta\}$ available.

Ideally, all of the base and enhancement layers are consumed throughout playback, i.e., the encoded video is sent to the client decoder at rate $r_b(t) + r_e(t)$ for all $0 \leq t \leq T$. Due to limited and fluctuating available bandwidth, however, it may not be possible to deliver all data to the client decoder by their deadline. Our goal is to identify the policies that minimize the loss in the base and enhancement layers.

IV. Infinite-Length Video

We first study the dynamic bandwidth allocation problem among layers for infinite-length video. The infinite-length case approximates the streaming of a full-length movie for which T is very large. Let $\bar{r}_b$ denote the average encoding rate of the base layer, that is,

$$\bar{r}_b = \lim_{T\to\infty} \frac{1}{T} \int_0^T r_b(t)\, dt.$$

Similarly, define $\bar{r}_e$ to be the average encoding rate of the enhancement layer. For the infinite video case, we assume that $\{X(t), t \geq -\Delta\}$ is a stationary and ergodic stochastic process. Let $\lambda = \mathrm{E}[X(t)]$ denote the (a priori unknown) average available bandwidth.

A. Loss Rates

Loss of data from the base layer can occur only when $Y_b(t) = 0$ and $\pi_b(t)X(t) < r_b(t)$. We make the natural assumption throughout this paper that when these two conditions occur, the data resulting from $\pi_b(t)X(t)$ can be used to approximate the decoded video stream. (This could be done, for example, by using an error concealment scheme such as replacing missing blocks of video with blocks from earlier frames.) The rate at which loss occurs when $Y_b(t) = 0$ is $[r_b(t) - \pi_b(t)X(t)]^+$. Thus the long-run fraction of base-layer traffic lost is

$$P_b^\pi = \lim_{T\to\infty} \frac{\int_0^T [r_b(t) - \pi_b(t)X(t)]^+ \mathbf{1}(Y_b(t) = 0)\, dt}{\int_0^T r_b(t)\, dt}.$$

P_b^π should be interpreted as the long-run fraction of the compressed video that is not consumed at the client. In a similar manner we define Q_e^π to be the long-run fraction of enhancement traffic lost:

$$Q_e^\pi = \lim_{T\to\infty} \frac{\int_0^T [r_e(t) - \pi_e(t)X(t)]^+ \mathbf{1}(Y_e(t) = 0)\, dt}{\int_0^T r_e(t)\, dt}.$$

Q_e^π is not an appropriate measure for the fraction of enhancement traffic that is *effectively* lost from the video stream. Recall that a critical property of layered video is that to decode the enhancement layer, the base layer must be available at the client. As a result, there is loss of enhancement traffic whenever there is loss of traffic from the base layer, even if $Y_e(t) > 0$. We first suppose that when there is "partial loss" of base-layer traffic, there is also "partial loss" of enhancement-layer traffic. In this partial-loss model, the fraction of encoded enhancement-layer traffic that is consumed can be as much as the fraction of encoded base-layer traffic consumed. This model

would be appropriate when many of the available blocks in the enhancement layer of a frame are blocks available in the base layer of the same frame. Note that the fraction of base-layer traffic that is consumed during base-layer loss is $\pi_b(t)X(t)/r_b(t)$. The partial loss model supposes that an equal fraction of enhancement-layer traffic is consumed in the case when $Y_e(t) > 0$ and there is loss of base-layer traffic. Thus, in that case, enhancement-layer traffic is consumed at rate $r_e(t) \cdot \pi_b(t)X(t)/r_b(t)$. More generally, we define the long-run fraction of enhancement-layer traffic effectively lost as

$$P_e^\pi = \lim_{T\to\infty} \frac{\int_0^T [r_e(t) - H(t)]^+ \, dt}{\int_0^T r_e(t)\, dt}, \tag{1}$$

where $H(t)$ is the consumption rate of enhancement-layer traffic at time t, i.e.,

$$H(t) = \begin{cases} r_e(t) & \text{when } Y_b(t) > 0, Y_e(t) > 0 \\ \pi_e(t)X(t) & \text{when } Y_b(t) > 0, Y_e(t) = 0 \\ r_e(t)\frac{\pi_b(t)X(t)}{r_b(t)} & \text{when } Y_b(t) = 0, Y_e(t) > 0 \\ \min\{\pi_e(t)X(t), r_e(t)\frac{\pi_b(t)X(t)}{r_b(t)}\} & \text{otherwise.} \end{cases} \tag{2}$$

B. Feasible Region

Having defined the loss probabilities P_b^π and P_e^π, we now identify the set of possible (P_b^π, P_e^π) values. We show that the loss probability tuple (P_b^π, P_e^π) belongs to a feasible set Ω, where Ω is the set of all tuples (P_b, P_e) that satisfy

$$P_e \geq P_b \tag{3}$$

$$\bar{r}_b(1 - P_b) + \bar{r}_e(1 - P_e) \leq \lambda \tag{4}$$

$$P_b \geq [1 - \frac{\lambda}{\bar{r}_b}]^+ \tag{5}$$

The feasible set Ω is shown in Fig. 3. Note that region $\mathcal{L}$ represents an upper bound on the performance level that can be achieved. The inequality (3) follows directly from the definitions of P_b^π and P_e^π. To prove (4), let $P_b^\pi(t)$ be the fraction of base-layer traffic lost over $[0, t]$ for a general prefetch policy π. Similarly, define $Q_e^\pi(t)$ for the enhancement layer. The amount of base-layer traffic that has been consumed up to time t is the amount of traffic that has been delivered to the client up to time t minus the amount of traffic that remains in the client prefetch buffer at time t. Thus, we have

$$1 - P_b^\pi(t) = \frac{\int_{-\Delta}^t \pi_b(s)X(s)\, ds - Y_b^\pi(t)}{\int_0^t r_b(s)\, ds} = \frac{\frac{1}{t}\int_{-\Delta}^t \pi_b(s)X(s)\, ds - \frac{1}{t}Y_b^\pi(t)}{\frac{1}{t}\int_0^t r_b(s)\, ds}$$

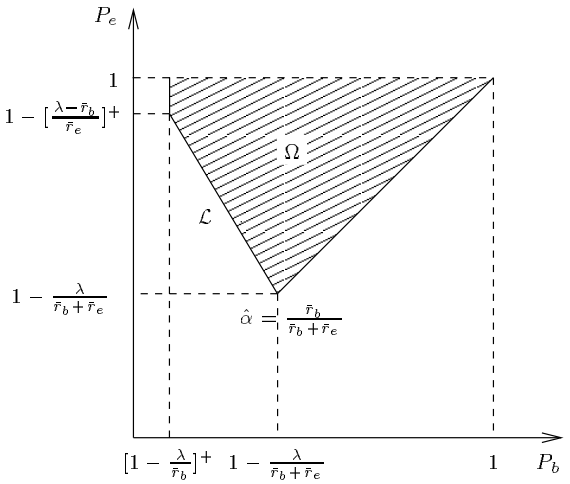


Fig. 3. Set of feasible loss probabilities

Similarly,

$$1 - Q_e^\pi(t) = \frac{\frac{1}{t}\int_{-\Delta}^t \pi_e(s)X(s)\, ds - \frac{1}{t}Y_e^\pi(t)}{\frac{1}{t}\int_0^t r_e(s)\, ds}$$

Combining the above two equations and using $\pi_b(t) + \pi_e(t) = 1$ gives

$$\frac{1}{t}\int_0^t r_b(s)ds\,[1 - P_b^\pi(t)] + \frac{1}{t}\int_0^t r_e(s)ds\,[1 - Q_e^\pi(t)] = \frac{1}{t}\int_{-\Delta}^t X(s)ds - \frac{1}{t}\left[Y_b^\pi(t) + Y_e^\pi(t)\right].$$

Taking the limit of the both sides of the above equation gives

$$\bar{r}_b(1 - P_b^\pi) + \bar{r}_e(1 - Q_e^\pi) = \lambda.$$

The proof of (4) is completed by noting that by definition $P_e^\pi \geq Q_e^\pi$. Relationship (5) follows from a similar argument and noting that P_b^π is minimized by setting $\pi_b(t) = 1$ for all t.

Having shown that all tuples (P_b^π, P_e^π) belong to Ω, which tuples in this region provide the best performance? The answer to this question depends on the relative importance of the base and enhancement layers, which in turn depends on the specific compression and error concealment schemes employed. It may be desirable to trade off small increases in base-layer loss for large decreases in effective enhancement layer loss, thereby improving overall image quality. In any case, tuples falling on $\mathcal{L}$ dominate tuples falling in $\Omega - \mathcal{L}$: for any point belonging to $\Omega - \mathcal{L}$, there exists points on $\mathcal{L}$ that provide strictly better performance. We therefore say that a policy π is *optimal* if (P_b^π, P_e^π) belongs to $\mathcal{L}$. In the following subsection we show that a very simple class of policies can achieve all the points on $\mathcal{L}$, attaining thereby optimal performance.

C. Optimality of Static Policies

In this subsection we consider a specific class of policies for which the allocation $\pi_b(t)$ is constant. Let α_b, $0 \le \alpha_b \le 1$, denote such a *static policy*. In the static allocation scheme, a constant fraction of the available bandwidth is allocated to each layer throughout the video transmission. Thus, the base-layer prefetch buffer is fed at rate $\alpha_b X(t)$ and the enhancement-layer prefetch buffer is fed at rate $\alpha_e X(t)$, where $\alpha_b + \alpha_e = 1$. Furthermore, define $\hat{\alpha} = \frac{\bar{r}_b}{\bar{r}_b + \bar{r}_e}$. Intuitively, the static policy $\hat{\alpha}$ allocates transmission rate to each layer in proportion to its long-run average consumption rate. A static policy is relatively easy to implement as it does not depend on prefetch buffer contents. The following theorem presents our first main result, namely, static policies are optimal for the infinite-video case. For this theorem, we assume that $r_b(t) = K r_e(t)$ for some constant K; this assumption trivially holds for the CBR case and is likely to roughly hold for the VBR case.

Theorem 1: Each point on the dominating region $\mathcal{L}$ is achieved by some static policy with $\hat{\alpha} \le \alpha_b \le \min\{1, \frac{\bar{r}_b}{\lambda}\}$.

Proof: Note that $\mathcal{L}$ is the boundary of feasible set Ω attained when (4) is binding (i.e., when it holds as an equality). By an argument similar to that in the proof of (4), it can be shown that

$$P_b^\alpha = [1 - \frac{\alpha_b \lambda}{\bar{r}_b}]^+, \tag{6}$$

and similarly,

$$Q_e^\alpha = [1 - \frac{\alpha_e \lambda}{\bar{r}_e}]^+. \tag{7}$$

It follows from the above two equations and from $\alpha_b + \alpha_e = 1$ that P_b^α and Q_e^α satisfy

$$\bar{r}_b(1 - P_b^\alpha) + \bar{r}_e(1 - Q_e^\alpha) = \lambda. \tag{8}$$

Furthermore, it follows from (6) that as α_b varies from 1 to $\hat{\alpha}$, P_b^α varies from $[1 - \frac{\lambda}{\bar{r}_b}]^+$ to $1 - \frac{\lambda}{\bar{r}_b + \bar{r}_e}$, where the last two values are the P_b values at the endpoints of $\mathcal{L}$. Thus, to prove that all points on $\mathcal{L}$ are attained by static prefetch policies with $\hat{\alpha} \le \alpha_b \le \min\{1, \frac{\bar{r}_b}{\lambda}\}$, it suffices to show that for this set of policies (4) is binding. Equation (8) implies that this is clearly the case when $P_e^\alpha = Q_e^\alpha$. To complete the proof of the theorem it thus suffices to show that for static policies with $\hat{\alpha} \le \alpha_b \le \min\{1, \frac{\bar{r}_b}{\lambda}\}$

$$P_e^\alpha = Q_e^\alpha \tag{9}$$

Fix a realization $\{x(t), t \ge -\Delta\}$ of stochastic process $X(t)$. Additionally, fix realizations $\{y_b(t), t \ge -\Delta\}$ and $\{y_e(t), t \ge -\Delta\}$ of the prefetch buffer content functions $Y_b(t)$ and $Y_e(t)$, respectively. First, we define the normalized buffer content functions as $\tilde{y}_b(t) = y_b(t)/\bar{r}_b$ and $\tilde{y}_e(t) = y_e(t)/\bar{r}_e$. Taking derivatives of $\tilde{y}_b(t)$ and $\tilde{y}_e(t)$ we obtain

$$\tilde{y}_b'(t) = \begin{cases} \frac{\alpha_b x(t) - r_b(t)}{\bar{r}_b} & \text{when } \tilde{y}_b(t) > 0 \\ \left[\frac{\alpha_b x(t) - r_b(t)}{\bar{r}_b}\right]^+ & \text{when } \tilde{y}_b(t) = 0 \end{cases} \tag{10}$$

and

$$\tilde{y}_e'(t) = \begin{cases} \frac{\alpha_e x(t) - r_e(t)}{\bar{r}_e} & \text{when } \tilde{y}_e(t) > 0 \\ \left[\frac{\alpha_e x(t) - r_e(t)}{\bar{r}_e}\right]^+ & \text{when } \tilde{y}_e(t) = 0 \end{cases} \tag{11}$$

By condition $\alpha_b \ge \hat{\alpha}$ and by noting that $r_b(t) = K r_e(t)$ implies $\bar{r}_b = K \bar{r}_e$, we obtain

$$\frac{\alpha_b x(t) - r_b(t)}{\bar{r}_b} \ge \frac{\alpha_e x(t) - r_e(t)}{\bar{r}_e} \quad \text{for all } t. \tag{12}$$

We first claim that

$$\tilde{y}_e(t) > 0 \text{ implies } \tilde{y}_b'(t) \ge \tilde{y}_e'(t). \tag{13}$$

To see this, note that $\tilde{y}_e(t) > 0$ implies

$$\begin{aligned} \tilde{y}_e'(t) &= \frac{\alpha_e x(t) - r_e(t)}{\bar{r}_e} \\ &\le \frac{\alpha_b x(t) - r_b(t)}{\bar{r}_b}, \end{aligned} \tag{14}$$

where the equality follows from (11) and the inequality follows from (12). Also from (10) we have

$$\tilde{y}_b'(t) \ge \frac{\alpha_b x(t) - r_b(t)}{\bar{r}_b}. \tag{15}$$

Combining (14) and (15) we establish (13). We now prove that

$$y_b(t) = 0 \text{ implies } y_e(t) = 0. \tag{16}$$

It suffices to show that

$$\tilde{y}_b(t) \ge \tilde{y}_e(t) \text{ for all } t. \tag{17}$$

Fix a $t \ge 0$. Clearly, (17) is true when $\tilde{y}_e(t) = 0$. Now suppose that $\tilde{y}_e(t) > 0$. Then t belongs to a busy period of $\tilde{y}_e(t)$. Let σ denote the starting time of the busy period of $\tilde{y}_e(t)$; we have $\tilde{y}_e(\sigma) = 0$. Furthermore $\tilde{y}_b(\sigma) \ge 0$. Thus at the beginning of the busy period, $\tilde{y}_b(\sigma) \ge \tilde{y}_e(\sigma)$. For all s within the busy period $\tilde{y}_b'(s) \ge \tilde{y}_e'(s)$ by (13). These two facts imply that $\tilde{y}_b(s) \ge \tilde{y}_e(s)$ for all s in the busy period, and in particular $\tilde{y}_b(t) \ge \tilde{y}_e(t)$, which establishes (17), and in turn implies (16).

We now complete the proof of (9). Recall that P_e^α is in general given by (1) and (2). By applying (16), (2) becomes

$$H(t) = \begin{cases} r_e(t) & \text{when } Y_e(t) > 0 \\ \alpha_e X(t) & \text{when } Y_e(t) = 0 \end{cases} \tag{18}$$

Note that for the case when $Y_b(t) = Y_e(t) = 0$ in (2), condition $\alpha_b \geq \hat{\alpha}$ implies that $\alpha_e \leq r_e(t) \cdot \alpha_b / r_b(t)$, and $H(t)$ reduces to $\alpha_e X(t)$. Using (18) in (1) yields $P_e^\alpha = Q_e^\alpha$ ■

Theorem 1 indicates that optimal performance is achieved by a static policy. The specific optimal policy $\alpha_b \in [\hat{\alpha}, \min\{1, \frac{\bar{r}_b}{\lambda}\}]$, however, depends on the relative importance of the base and enhancement layers. As an example, suppose that user perceived quality is maximized by making P_b^π as small as possible. In that case, the optimal policy is to set $\alpha_b = \min\{1, \bar{r}_b/\lambda\}$. To implement this policy, the server does not need to keep track of the prefetch buffer contents. The server must, however, have an estimate of the average available bandwidth λ. At any time t, such an estimate can be based on the current available bandwidth $X(t)$ and all its past values. For example, the server can dynamically estimate λ at time t as follows:

$$\lambda = \frac{\int_{-\Delta}^{t} e^{-\mu(t-s)} X(s)\, ds}{\int_{-\Delta}^{t} e^{-\mu(t-s)}\, ds}, \tag{19}$$

for some damping factor μ. Given the most recent estimate for λ , the server can then adjust the optimal value of α_b. Note, finally, that in the case when λ exceeds the total average consumption rate $\bar{r}_b + \bar{r}_e$, then a reasonable policy is $\alpha_b = \hat{\alpha}$, regardless of the relative importance of the layers.

D. Total Loss Model

Our analysis of the bandwidth allocation problem for the case of infinite-length video in the previous subsections has been based on the assumption that during instants of base-layer traffic loss, an equal fraction of enhancement-layer traffic is lost, even if $Y_e(t) > 0$. We referred to the above as the partial-loss model. We now consider a second model for enhancement-layer loss in which *no* encoded enhancement-layer traffic can be consumed when there is loss of encoded base-layer traffic. We refer to this model as the *total-loss model*. Note that this model still permits partial decoding of the enhancement layer when *all* of the base layer is available. In this subsection we determine the loss rates and the optimal streaming policy for this second model. In the total-loss model, enhancement-layer traffic is lost at rate $r_e(t)$ when $Y_b(t) = 0$ and $\pi_b(t)X(t) < r_b(t)$. The long-run fraction of enhancement-layer traffic that is effectively lost is given by (1), where the effective consumption rate $H(t)$ when $Y_b(t) > 0$ or $r_b(t) \leq \pi_b(t)X(t)$ is given by

$$H(t) = \begin{cases} r_e(t) & \text{if } Y_e(t) > 0 \\ \pi_e(t)X(t) & \text{if } Y_e(t) = 0 \end{cases}$$

and by $H(t) = 0$, when $Y_b(t) = 0$ and $r_b(t) > \pi_b(t)X(t)$.

Let R_e denote the long-run fraction of enhancement-layer traffic lost for the total-loss model. Naturally, the optimal policy will favor more the base layer, as 100% of enhancement-layer traffic is lost even if only a small fraction of traffic is lost from the base layer. As with the partial-loss model, the optimal streaming policy must ensure that the enhancement-layer prefetch buffer is empty whenever there is loss in the base layer. Additionally, due to the total-loss assumption, no enhancement traffic should be streamed during times when there is loss in the base layer. Using the techniques of subsection IV-B, it can be shown that the tuple (P_b, R_e) belongs to the feasible set Ω as defined by equations (3)-(5). Now consider policy $\pi^\alpha = (\pi_b^\alpha(t), t \geq -\Delta)$, which we define as follows:

$$\pi_b^\alpha(t) = \begin{cases} \alpha_b & \text{if } Y_b(t) > 0 \text{ or } r_b(t) \leq \alpha_b X(t) \\ \min\{1, \frac{r_b(t)}{X(t)}\} & \text{otherwise.} \end{cases}$$

Policy π^α allocates a constant fraction α_b of the bandwidth to the base layer when either $Y_b(t) > 0$ or when the current allocation exceeds the current consumption rate. When the base-layer prefetch buffer is empty, policy π^α may increase the fraction of bandwidth allocated to the base layer to avoid loss of base-layer traffic. This is done by either allocating to the base layer a fraction of the available bandwidth equal to $\frac{r_b(t)}{X(t)}$ if $r_b(t) \leq X(t)$, or by allocating to the base-layer all of the available bandwidth if $r_b(t) > X(t)$. In the former case, the allocation avoids base-layer loss, but does not prefetch any enhancement-layer traffic. Note that policy π^α allocates no bandwidth to the enhancement layer unless $Y_b(t) > 0$. It can be shown in a manner similar to the proof of Theorem 1, that under policy π^α with $\alpha_b > \hat{\alpha}$, $Y_b(t) = 0$ implies $Y_e(t) = 0$. This relationship in turn implies that (P_b, R_e) tuples for policy π^α with $\alpha_b > \hat{\alpha}$ belong to region $\mathcal{L}$, as indicated in Fig. 3, i.e.,

$$\bar{r}_b(1 - P_b) + \bar{r}_e(1 - R_e) = \lambda$$

The above relationship can be shown again by using similar arguments as in the proof of Theorem 1. Consequently, (P_b, R_e) tuples for policies π^α with $\alpha_b > \hat{\alpha}$ dominate all other points in Ω, thereby achieving optimality.

V. FINITE-LENGTH VIDEO

In this section we consider the layered prefetching problem for the case of finite-length video. The finite-length case models the situation in which a short clip (i.e., T is relatively small) is to be streamed from server to client. In this analysis, we again consider VBR-encoded video, although our results remain valid for the special case of CBR-encoded video.

Recall that $\pi = (\pi_b(t), t \geq -\Delta)$ denotes a general streaming policy, where $\pi_b(t)$ is the fraction of $X(t)$ allocated to the base layer at time t and $\pi_e(t) = 1 - \pi_b(t)$ is the fraction of $X(t)$ allocated to the enhancement layer. For the finite-length case, we need to restrict the general streaming policy π so that as soon as the streaming of a layer is complete, the total available transmission rate is allocated to the layer for which data remains to be sent. To this purpose, we define parameters T_b and T_e that indicate the times at which the streaming of each layer is complete. At T_b, for instance, the portion of base-layer data that remains to be consumed up through time T has been downloaded into the prefetch buffer. Specifically, we define

$$T_b = \min\{t : Y_b(t) = \int_t^T r_b(s)\, ds\}.$$

We define an analogous expression for T_e. Note that in the finite-length case, the bandwidth that is available between $\max\{T_b, T_e\}$ and T is not utilized. Let P_b^π be the fraction of base-layer traffic lost. Furthermore, let $T_{\min} = \min\{T_b, T_e\}$. P_b^π is given by

$$P_b^\pi = \frac{\int_0^T [r_b(t) - H_b(t)]^+\, dt}{\int_0^T r_b(t)\, dt}, \quad \text{where}$$

$$H_b(t) = \begin{cases} \pi_b(t)X(t) & \text{when } Y_b(t) = 0 \text{ for } t < T_{\min} \\ X(t) & \text{when } Y_b(t) = 0 \text{ for } T_{\min} \leq t \leq T_b \\ r_b(t) & \text{otherwise.} \end{cases}$$

Similarly,

$$Q_e^\pi = \frac{\int_0^T [r_e(t) - H_e(t)]^+\, dt}{\int_0^T r_e(t)\, dt}, \quad \text{where}$$

$$H_e(t) = \begin{cases} \pi_e(t)X(t) & \text{when } Y_e(t) = 0 \text{ for } t < T_{\min} \\ X(t) & \text{when } Y_e(t) = 0 \text{ for } T_{\min} \leq t \leq T_e \\ r_e(t) & \text{otherwise.} \end{cases}$$

Clearly, loss of base-layer traffic is only possible for $t \leq T_b$. Note, however, that loss of enhancement-layer traffic is possible for $t \geq T_e$. As Q_e^π does not represent the actual loss in the enhancement layer, we next determine the fraction of enhancement-layer traffic P_e^π effectively lost according to the partial loss model. Recall that in the partial loss model, the fraction of traffic lost from the enhancement layer when $Y_e > 0$ and there is loss of data in the base layer equals the fraction of traffic lost from the base layer. Specifically, the fraction of enhancement traffic effectively lost is

$$P_e^\pi = \frac{\int_0^T [r_e(t) - H(t)]^+\, dt}{\int_0^T r_e(t)\, dt}, \tag{20}$$

where

$$H(t) = \min\{\frac{r_e(t)}{r_b(t)} H_b(t), H_e(t)\}. \tag{21}$$

The above expression for the effective consumption rate in the enhancement layer follows directly from the definitions of the loss rates in the partial-loss model. Note that these definitions imply that for any policy π

$$P_e^\pi \geq \max\{P_b^\pi, Q_e^\pi\}. \tag{22}$$

A. Preliminary Results

Having defined the loss probabilities for the streaming of finite-length video, we now present some necessary preliminary results, which will aid in the derivation of the optimal streaming policies. For detailed proofs of these results see [23]. For these results, we again assume that $r_b(t) = K r_e(t)$ for some constant K. Let $\bar{r}_b$ denote the average encoded rate of the base layer, i.e., $\bar{r}_b = \frac{1}{T}\int_0^T r_b(t)\, dt$. Similarly define $\bar{r}_e$. We consider the class of static policies and establish the following lemma, which parallels the results obtained in Section IV-C for the infinite-length case.

Lemma 1: (a) Fix a static policy α_b. If $\alpha_b \geq \hat{\alpha}$, then (i) $Y_b(t) = 0$ implies $Y_e(t) = 0$; (ii) $T_b^\alpha \leq T_e^\alpha$;
(iii) $P_e^\alpha = Q_e^\alpha$.
(b) If $\alpha_b \leq \hat{\alpha}$, then (i) $Y_e(t) = 0$ implies $Y_b(t) = 0$; (ii) $T_b^\alpha \geq T_e^\alpha$; (iii) $P_e^\alpha = P_b^\alpha$.

From Lemma 1, we have $T_e^{\hat{\alpha}} = T_b^{\hat{\alpha}}$,i.e., under policy $\hat{\alpha}$ streaming for both layers ends at the same time. To simplify notation, write T_c for $T_e^{\hat{\alpha}}$. We next present a second important result, which establishes a key property of the static policy $\hat{\alpha}$.

Lemma 2: $\max\{T_b^\pi, T_e^\pi\} \leq T_c$ for any policy π.
Lemma 2 states that policy $\hat{\alpha}$ maximizes the streaming duration for both layers, thereby utilizing available bandwidth for at least as long as any policy π. As we shall see, in the case when both layers are equally important, this property is key in achieving optimality.

B. Optimization Problem

We now use the results in the previous subsection to determine the optimal streaming policy. We approach this problem by formulating and solving the following optimization problem:

$$\max_\pi \quad J_\pi = \mathrm{E}[d_b(1 - P_b^\pi) + d_e(1 - P_e^\pi)], \tag{23}$$

in which d_b and d_e are fixed constants and denote the relative importance of the encoded base and enhancement layers. Note that when $d_b = \bar{r}_b$ and $d_e = \bar{r}_e$, we are optimizing the expected sum of base and enhancement layer average throughputs. Throughout this section we suppose

$$\frac{d_e}{d_b} \geq \frac{\bar{r}_e}{\bar{r}_b}. \tag{24}$$

Condition (24) implies that the enhancement layer has a greater (or equal) impact than the base layer on the quality of the decoded video. We thus seek the optimal streaming policy π that maximizes the expected weighted fraction of traffic consumed in both layers, for the case when the enhancement layer is considered at least as important as the base layer. For this case, we shall show that the static policy $\hat{\alpha} = \frac{\bar{r}_b}{\bar{r}_b+\bar{r}_e}$ achieves optimality. We consider the optimization of the same objective function when the encoded base-layer stream has a greater impact on quality than the enhancement-layer stream in a following subsection.

We approach the optimization problem in (23)-(24) by first solving the simpler problem

$$\max_{\pi} \quad F_\pi = \mathrm{E}\left[\bar{r}_b(1-P_b^\pi) + \bar{r}_e(1-P_e^\pi)\right],$$

and then showing that the obtained solution is also optimal for (23)-(24). The following theorem states that policy $\hat{\alpha}$ optimizes function F_π.

Theorem 2: The policy $\hat{\alpha}$ is optimal for F_π, i.e., $F_{\hat{\alpha}} \geq F_\pi$, for any policy π.

Proof: Using the results of Lemma 1, and the definitions for the loss probabilities, it can be shown that

$$F_{\hat{\alpha}} = \mathrm{E}\left[\frac{1}{T}\int_0^{T_c} X(t)\,dt\right].$$

Additionally, it can be shown that

$$F_\pi \leq \mathrm{E}\left[\frac{1}{T}\int_0^{\max\{T_b^\pi, T_e^\pi\}} X(t)dt\right].$$

Applying Lemma 2 to the right-hand side of the above two relationships yields

$$\mathrm{E}\left[\frac{1}{T}\int_0^{T_c} X(t)\,dt\right] \geq \mathrm{E}\left[\frac{1}{T}\int_0^{\max\{T_b^\pi, T_e^\pi\}} X(t)dt\right],$$

which implies that $F_{\hat{\alpha}} \geq F_\pi$ ■

We now turn to the maximization problem in (23)-(24). Using Theorem 2 we derive the following (see [23]).

Corollary 1: The policy $\hat{\alpha}$ is optimal for J_π, i.e., $J_{\hat{\alpha}} \geq J_\pi$ for any π when $\frac{d_e}{d_b} \geq \frac{\bar{r}_e}{\bar{r}_b}$.

The above corollary states that when the enhancement layer has an equal effect on the quality of the decoded video as the base layer, the optimal policy is a specific static policy, namely, the optimal policy allocates a constant fraction of bandwidth to each layer in proportion to each layer's transmission rate. Again, the optimal policy utilizes prefetching when possible, but is independent of the prefetch buffer contents.

VI. Heuristics for Finite-Length Video

Having solved the layered streaming problem for the case when the enhancement layer has a relatively high impact on quality, we now consider the important case in which the base layer has a greater impact on quality than the enhancement layer. We suppose throughout this section that

$$\frac{d_e}{d_b} < \frac{\bar{r}_e}{\bar{r}_b}. \tag{25}$$

Under this condition , the complexity of the optimization problem in (23) increases significantly. In this paper, we do not provide an analytical solution to the optimal streaming problem under condition (25). Instead, we develop heuristic streaming policies and investigate the performance of these policies through a simulation study. Our results show that static policies can perform poorly when (25) holds.

A. Bounds on Performance

There are upper bounds on the best possible performance that can be achieved by any streaming policy. In this subsection we derive two types of upper performance bounds. We will later compare these bounds to the performance of our heuristic streaming policies for finite-length video. A first bound results from a traffic conservation relationship. It can be shown using a simple traffic conservation statement (see [23]) that the following holds

$$\bar{r}_b(1-P_b^\pi) + \bar{r}_e(1-Q_e^\pi) = \frac{1}{T}\int_{-\Delta}^{\max\{T_b^\pi, T_e^\pi\}} X(t)\,dt.$$

Using $P_e^\pi \geq \max\{P_b^\pi, Q_e^\pi\}$ and $T_c \geq \max\{T_b^\pi, T_e^\pi\}$ (from Lemma 2), and taking the expectation of both sides gives the bound

$$\bar{r}_b(1-\mathrm{E}[P_b^\pi]) + \bar{r}_e(1-\mathrm{E}[P_e^\pi]) \leq C_1, \tag{26}$$

where

$$C_1 = \frac{1}{T}\mathrm{E}[\int_{-\Delta}^{T_c} X(t)\,dt]. \tag{27}$$

Note that C_1 in this bound is a constant and does not depend on policy π.

A second performance bound can be obtained by noting that loss in the base layer is always minimized when all of the available bandwidth is allocated to the base layer,

until this layer is fully prefetched. Applying static policy $\alpha_b = 1$ (see [23]) yields

$$\mathrm{E}[P_b^{\pi}] \geq 1 - \frac{\mathrm{E}[\int_{-\Delta}^{T_b^1} X(t)\,dt]}{\int_0^T r_b(t)\,dt} = C_2, \qquad (28)$$

where T_b^1 is T_b^{α} with $\alpha = 1$. Note that C_2 does not depend on π.

B. Threshold Policies and Simulation

We now consider heuristic streaming policies for finite-length video. We begin by defining a heuristic threshold policy denoted by $\hat{\pi}$, which varies the fraction of bandwidth allocated to each layer according to current prefetch buffer contents. In particular, when the content of the base layer prefetch buffer is below a certain constant threshold, denoted by q_{thres}, policy $\hat{\pi}$ allocates all of the available bandwidth to the base layer. When the base layer prefetch buffer content exceeds the threshold, policy $\hat{\pi}$ decreases the fraction of the bandwidth allocated to the base layer to $\hat{\alpha}$. Once the base layer has been entirely prefetched, the policy allocates all available bandwidth to the enhancement layer. Thus $\hat{\pi} = (\pi_b(t), t \geq -\Delta)$, where $\pi_b(t)$ at time t is given by

$$\pi_b(t) = \begin{cases} 1 & \text{when } Y_b(t) < q_{\text{thres}} \\ \hat{\alpha} & \text{when } Y_b(t) \geq q_{\text{thres}} \\ 0 & \text{when } Y_b(t) > \int_t^T r_b(s)\,ds \end{cases}$$

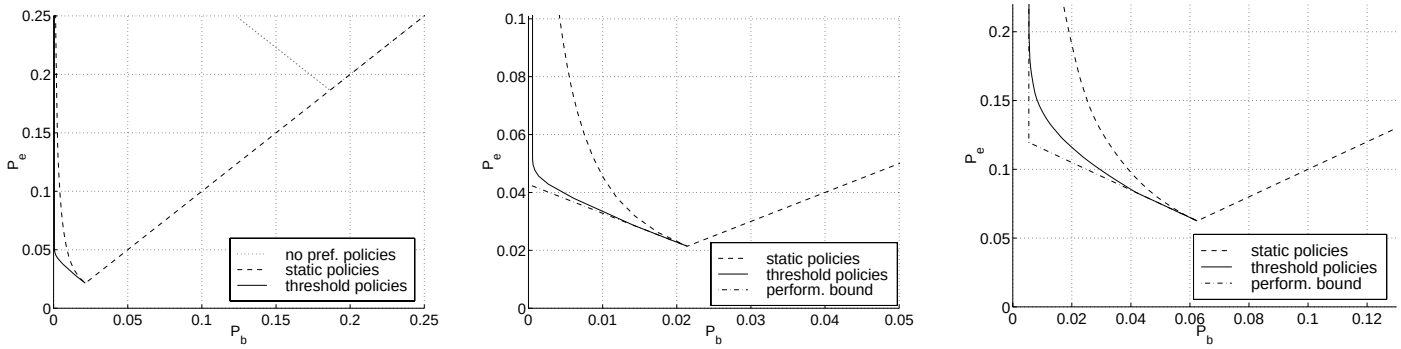
A key issue in the implementation of the threshold policy is making a reasonable choice for the value of the threshold. High threshold values may lead to overly conservative policies that result in unacceptable losses in the enhancement layer for insignificant improvement in the base layer losses. On the other hand, very low thresholds may result in unsatisfactory performance in terms of the losses incurred in the base layer. The development of heuristics for determining appropriate threshold values is an area of ongoing work.

We have investigated the performance of a number of streaming policies, including the threshold policies defined above, in a simulation study. In this study we used a specific stochastic model for $X(t)$. Specifically, we let $X(t)$ vary randomly among two constant levels C_1 and C_2, with probability p and $(1-p)$, respectively. We let $X(t)$ remain in each of the two states for a random period of time. We denote τ_1 and τ_2 for the mean duration in each state of $X(t)$. Note that $p = \frac{\tau_1}{\tau_1+\tau_2}$. We define the system utilization by $\rho = \frac{\mathrm{E}[X(t)]}{(\bar{r}_b+\bar{r}_e)}$.

Fig. 4 shows the results of a simulation study in which $\rho = 1$. Each of the three graphs plots the expected loss probability in the enhancement layer versus the expected loss probability in the base layer for two classes of streaming policies: static policies and threshold policies . Different static policies were evaluated by varying the value of α_b and different threshold policies were evaluated by varying the value of q_{thres}. Note that graph (a) also includes an additional class of streaming policies, namely, policies that do not employ prefetching. Graph (b) on the right simply represents a zoomed-in version of graph (a). Graph (c) was obtained by varying τ_1/T, while maintaining all other critical parameters such as ρ, p and T constant. Note that as τ_1/T increases, the likelihood of entering a long period during which $X(t)$ remains in the same state also increases. As we shall see, the existence of long periods during which $X(t)$ is constant has adverse consequences on performance. In graph (a), $\tau_1/T = 0.01$. The results illustrate that no-prefetching policies result in poor performance. For policies that employ no prefetching, (P_b, P_e) tuples are always dominated by (P_b, P_e) tuples resulting from static or threshold policies. This result confirms the significant benefits of prefetching. We see that the static policy $\hat{\alpha}$, which allocates bandwidth to the layers in proportion to their consumption rates, minimizes P_e for all cases. This is consistent with the results in subsection V-B, where it was shown that policy $\hat{\alpha}$ is optimal when the enhancement layer is at least as important as the the base layer. Graph (a) also illustrates the performance of threshold policies for different q_{thres} values. Clearly, when base-layer loss must be minimized, threshold policies attain significantly better performance than static policies.

The improvements attained by threshold policies are better seen in the zoomed in version of graph (a) on the right. A threshold policy resulting in expected base-layer loss of 0.5% gives a P_e near 4%. A static policy with a P_b of 0.5%, however, results in a P_e greater than 8.5%. Note that the zoomed-in version includes the upper performance bounds derived in section VI-A. The diagonal bound in the graph represents the bound in (26) obtained from the traffic conservation statement. The vertical bound in the graph indicates the minimum expected loss in the base layer determined by (28). As illustrated by the graph, the performance of threshold policies approximates the performance of the two bounds combined.

Graph (c) was obtained by setting $\tau_1/T = 0.1$. Increasing the value of τ_1/T has a negative effect on the performance of static and threshold policies, as seen in graph (b). A higher τ_1/T increases the likelihood of situations in which there are sustained periods of insufficient bandwidth. During these periods, video can not be prefetched and losses often become unavoidable. See [23] for additional numerical results.



(a) $\frac{\tau_1}{T} = 0.01$ (b) Magnified area of graph (a) (c) $\frac{\tau_1}{T} = 0.1$

Fig. 4. (P_b, P_e) tuples for three types of streaming policies: no-prefetching polices, static policies and threshold policies.

Note that the heuristic threshold policy $\hat{\pi}$ relies on a constant threshold level for the content of the base-layer prefetch buffer. A natural extension of the threshold policy is to utilize a dynamic threshold level. We are currently studying dynamic threshold policies and have a simple conservative estimate for the threshold value, which depends on the future base-layer consumption rate and on dynamic estimates for the future available bandwidth [23].

REFERENCES

[1] Real Networks, "RealSystem G2 Overview, RealNetworks Web Site," http://www.real.com/devzone/library/whitepapers/g2overview.html.

[2] Microsoft Corp., "Windows Media Player, Microsoft Web Site," http://www.microsoft.com/windows/mediaplayer/default.asp.

[3] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)," Tech. Rep., April 1998, RFC 2326.

[4] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," Tech. Rep., Jan. 1996, RFC 1889.

[5] D. Saparilla and Ross K. W., "Long-Time Scale TCP Throughput," http://www.eurecom.fr/ saparill, July 1999.

[6] J. Mahdavi and S. Floyd, "TCP-Friendly Unicast Rate-Based Flow Control," Tech. Rep., Jan. 1997.

[7] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of TCP congestion avoidance algorithm," *Computer Communications Review*, vol. 27, July 1997.

[8] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," in *Proc. of ACM SIGCOMM*, Sept. 1998.

[9] Y. Wang and Q. Zhu, "Error Control and Concealment for Video Communications: A Review," in *Proc. of the IEEE*, vol. 86, pp. 974 – 997. May 1998.

[10] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control*, Prentice-Hall, 1986.

[11] M. Reisslein and K. W. Ross, "High-Performance Prefetching Protocols for VBR Prerecorded Video ," in *IEEE Networking Magazine*. Nov./Dec. 1998.

[12] J. McManus and K. W. Ross, "A Dynamic Programming Methodology for Managing Prerecorded VBR Sources in Packet-Switched Networks," *Telecommunications Systems*, vol. 9, 1998.

[13] W. Feng, F. Jahanian, and S. Sechrest, "An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Prerecorded Video ," *ACM/Springer-Verlag Multimedia Systems*, vol. 5, no. 5, pp. 297–309, Sept. 1997.

[14] W. Feng and J. Rexford, "A Comparison of Bandwidth Smoothing Techniques for the Transmission of Prerecorded Compressed Video," in *Proc. of IEEE INFOCOM*, Kobe, Japan, Apr. 1997, pp. 58–66.

[15] J. M. McManus and K. W. Ross, "Video on Demand over ATM: Constant-Rate Transmission and Transport ," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1087–1098, 1996.

[16] J. Rexford, S. Sen, J. Dey, W. Feng, J. Kurose, J. Stankovic, and D. Towsley, "Online Smoothing of Live Variable-Bit-Rate Video," in *7th Workshop Network and Op. Systems Support for Digital Audio and Video*, St. Louis, MO, May 1997, pp. 249–257.

[17] Z.-L. Zhang, J. Kurose, J. Salehi, and D. Towsley, "Smoothing, Statistical Multiplexing and Call Admission Control for Stored Video," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 1148–1166, Aug. 1997.

[18] V. Jacobson, "Congestion Avoidance and Control," in *Proc. of ACM SIGCOMM*, Aug. 1988, pp. 314–329.

[19] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," Tech. Rep., Feb. 1998.

[20] R. Rejaie, M. Handley, and D. Estrin, "Architectural Considerations for Playback of Quality Adaptive Video over the Internet," Tech. Rep., USC, Nov. 1998.

[21] R. Rejaie, D. Estrin, and M. Handley, "Quality Adaptation for Congestion Controlled Video Playback over the Internet," in *To appear in Proc. of ACM SIGCOMM '99*, Cambridge, Sept. 1999.

[22] M. Podolsky, M. Vetterli, and S. McCanne, "Limited Retransmission of Real-time Layered Multimedia," in *IEEE Signal Processing Society*, L. Angeles, CA, Dec. 1998.

[23] D. Saparilla and Ross K. W., "Optimal Streaming of Layered Video," http://www.eurecom.fr/ saparill, July 1999.

[24] J. D. Salehi, Z. L. Zhang, J. Kurose, and D. Towsley, "Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing," *IEEE/ACM Transactions on Networking*, pp. 397–410, Aug. 1998.

[25] M. W. Garret and Fernandez A., "Variable Bit Rate Trace Using MPEG Code," Nov. 1994, Available at the ftp site: thumper.bellcore.com/pub/vbr.video.trace/MPEG.description.