Optimized Scalable Video Delivery for Streaming, Downloading, and Hybrid Access Models

S. Amir Hosseini, Fraida Fund, Shivendra S. Panwar
Department of Electrical and Computer Engineering
Polytechnic Institute of NYU
{amirhs.hosseini,ffund}@nyu.edu, panwar@catt.poly.edu

***Abstract*—In this work, we address the problem of scalable video delivery to mobile users under access constraints imposed by the content provider. Typically, these constraints dictate how much video data the consumer may cache on a local disk ahead of the video being played, and have a marked impact on the optimal policy for retrieving content. We consider three access models: streaming, downloading, and a hybrid access model called *streamloading*. Using a semi-Markov decision process (SMDP) we determine optimal video delivery policies for each of these access models and describe some high-level rules derived from these results.**

I. INTRODUCTION

A great deal of research energy has been focused on the challenge of delivering high-quality video content to mobile users, most of which exploits the idea of adaptive video. In adaptive video, the video is divided into segments, and multiple versions of each segment are created (encoded at different bit rates). When the next segment is to be downloaded for viewing (i.e., at a *decision epoch*), a decision is made (often based on conditions in the network or on the user's device) regarding which version of the segment to retrieve. The sequence of decisions form a *policy*, which can be designed to optimize any of a variety of metrics including video quality, network capacity, energy consumption, or fairness.

Some video delivery systems use scalable video coding (SVC), an extension of the H.264 video coding standard. In SVC, rather than encoding each segment into multiple bit rates, the video segments are encoded into layers. The *base layer* may be decoded into a low-quality video, and successive *enhancement layers* add incremental improvements in quality. In an adaptive scalable video scenario, the policy describes how many layers to download for each video segment, and in what sequence they should be downloaded.

There exists a substantial body of work on policies for delivery of video content to mobile users under various network, computing, and energy constraints. However, most of this work fails to consider the access constraints imposed by the content providers.

Video services in use today are generally *conditional access systems*. These systems enforce that a video is viewed according to rules that are stipulated by the content provider. For example, many Internet videos are available only to viewers in certain countries, viewers using specific operating systems, or viewers who have paid for access.

One common access rule dictates how much video data consumers may store on their local disk. In video *downloading* services this is not constrained, but in video *streaming* services, the consumer is not allowed to cache more than a short period of video data ahead of the point being watched. Content providers tend to prefer streaming services because it gives them fine-grained control over who may watch a video at any moment in time, as well as the ability to inject personalized advertisements into a video stream on-the-fly. For this reason, streaming services tend to be supported by advertisements or inexpensive, while downloading services are priced ten to a hundred times higher. However, streaming services tend to offer an inferior user experience for mobile users because the video quality is affected by fluctuations in wireless signal quality, network load, and other conditions in the access network.

In [1], we proposed a novel scalable video delivery service called *streamloading* that allows users to enjoy video quality similar to a downloading service, while still being legally classified as a streaming service from the content provider's point of view. In streamloading, the base layer of the video is streamed in real time, and enhancement layers may be downloaded ahead of time. The video quality is improved because the receiver can take advantage of available bandwidth to download enhancement layers of future segments, thereby reducing the effect of variations in link quality. However, the legal requirements of streaming are satisfied because the enhancement layers for future segments that are downloaded to the device cannot be played back without the base layer.

In this work, we use a semi-Markov decision process (SMDP) to derive a policy that optimizes video quality for scalable video delivery under three kinds of access rules: streaming, downloading, and streamloading. Because of the constraints imposed by these access rules, the policy tends to be very different in each case. We apply machine learning techniques to the data generated by the SMDP to find approximately optimal policies and codify some general rules for downloading scalable video in different scenarios.

II. METHODOLOGY

The form of the SMDP is as follows. The video is divided into segments, each having a playback duration of ten seconds. Each segment is encoded into one base layer and two enhancement layers. The *action* prescribed by the decision process is in

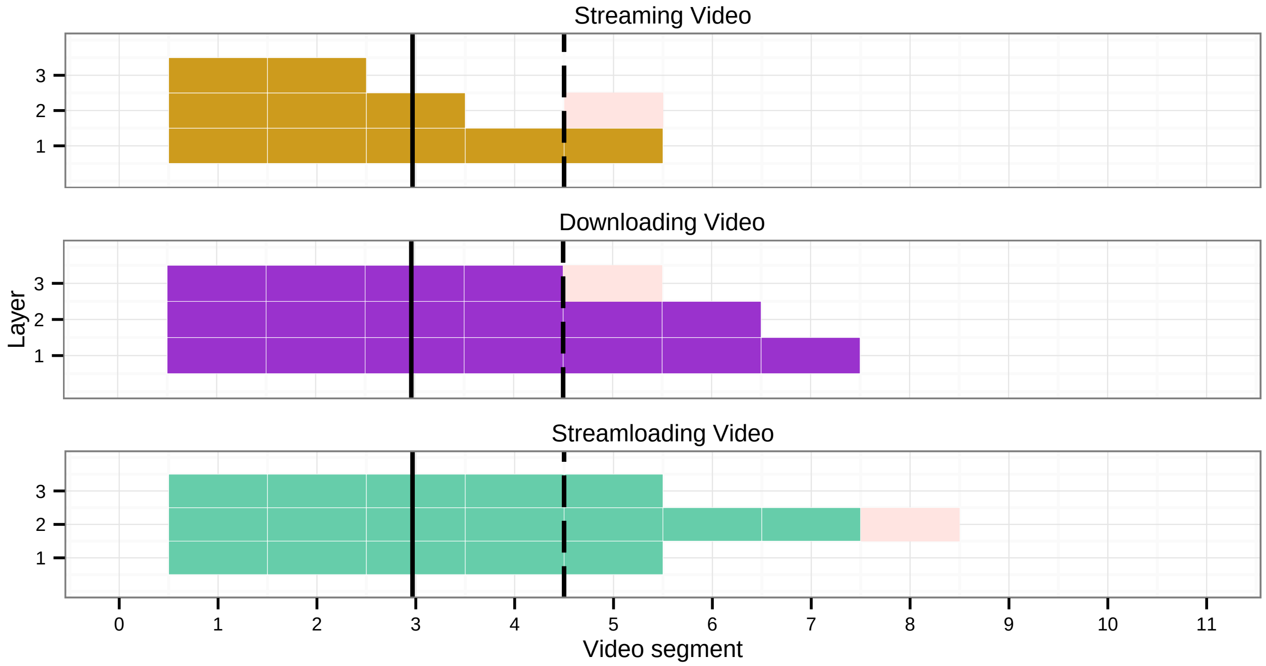

Fig. 1: A frame of the demo animation shows the SMDP at one decision epoch. Downloaded segments are represented by their sequence in the video and their layer index. The solid vertical line indicates the position of the playback header in the video, and the dashed vertical line gives the position of a download header. Video segments between these lines show the state of the playout buffer, while video segments to the right of the download header show the state of the download buffer. The current download at this decision epoch is colored light pink.

the form of a layer index (0,1, or 2) which indicates what layer of a future segment to download next. However, depending on the state and the access rule (streaming, downloading, or streamloading), only a subset of these actions are allowed.

The *state* of the process is represented by the content of the *download buffer*, the content of the *playout buffer*, the position of the *playback header*, and the channel quality. The download buffer is represented by three values, each indicating how many future segments have been retrieved for each of the three layers. Video segments that are being decoded in preparation for playback in less than four seconds are in the playout buffer, which is described by two values representing the number of layers of video in the first and second playback buffer positions. The playback header indicates the progress of video playback in the playout buffer.

The traversal from one state to another and the time spent in each state are based on the action taken by the SMDP and based on a random process representing the variation in channel quality. At each state, a reward $r_{s,a}$ is assigned for every possible action as follows:

$$r_{s,a} = \sum_{t=1}^{t_{s,a}} \log\left(\sum_{k=1}^{u_t} l_k\right), \qquad (1)$$

where $t_{s,a}$ is the duration of action a taken in state s, t represents a discretized interval of time spent in the state (up to $t_{s,a}$), u_t is the number of layers decoded for the video played back in timeslot t, and l_k is the size (in bits) of the kth layer of the video being played back in one time slot. To solve the SMDP we apply a value iteration method [2].

III. DEMO

Our demo shows how the policy determined by the MDP is applied for a given sequence of channel quality conditions, under each of the three access models (streaming, downloading, and streamloading). We play an animation of the policy (described in Figure 1) determined by the MDP for retrieving video segments in the streaming, downloading, and streamloading access modes. We also play back the video sequence decoded by the consumer for each of these modes.

REFERENCES

[1] A. Rath, S. Goyal, and S. Panwar, "Streamloading: low cost high quality video streaming for mobile users," in *Proceedings of the 5th Workshop on Mobile Video (MoVid '13)*. New York, NY, USA: ACM, 2013, pp. 1–6.

[2] P. J. Schweitzer, "Iterative solution of the functional equations of undiscounted markov renewal programming," *Journal of Mathematical Analysis and Applications*, vol. 34, no. 3, pp. 495–501, 1971.