The Effect of Channel Variations on Optimal Download Policies in SVC video delivery with buffer restrictions

S. Amir Hosseini, Fraida Fund, Shivendra S. Panwar
Department of Electrical and Computer Engineering
NYU Polytechnic School of Engineering
{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 varying buffering constraints on the user side to determine how channel variations affect the optimal delivery policy. Typically, buffering constraints dictate how much video data the consumer may cache on a local disk ahead of the playback instant, and have a marked impact on the optimal policy for retrieving content. Using a semi-Markov decision process (SMDP) we determine optimal video delivery policies for different levels of buffer limits in different wireless environments for a single user scenario. The outcome of the SMDP shows how the optimal download policy depends on the buffer constraints and the temporal correlation of the wireless channel. We apply a decision tree classifier to the output of the SMDP to derive simple approximate policies for different scenarios. The results indicate that the occupancy of the download buffer is in general more important in the decision making than the current state of the wireless channel. Furthermore, we show that the optimal policy is more conservative in slowly varying, and more greedy in fast changing channels.**

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. This offers a flexibility advantage over non-scalable adaptive video systems, in which once a quality decision is made for a particular segment, there is no possibility of incrementally increasing its quality by retrieving more data later. 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.

Within the context we have just described, 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, this body of work fails to consider the effect of channel variations on the optimal policy. It is argued in [**?**] that the temporal correlation of the channel capacity significantly impacts the delivered video quality. The unanswered question is if the optimal download policy for SVC video would change in different environments with different capacity variation characteristics.

Another important aspect in on demand video delivery is how much buffering is allowed on the end user device. A large buffer gives the content providers 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. It also results in a higher spectral efficiency for the video transmission. However, if users stop watching the video before it ends, the resources used for buffering the video will be wasted. Furthermore, a small buffer limit results in 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. Hence, based on the conditions and type of the video to be transmitted, content providers might choose varying amounts of buffer limits.

The goal of this work, therefore, is to determine the effect of different channel characteristics on the optimal video delivery policy for scalable video content under different buffering constraints. We use a semi-Markov decision process (SMDP) to find a policy that optimizes video quality for scalable video delivery under different access constraints, which in turn are defined as the amount of video being downloaded ahead of the playback time. We apply a decision tree classifier to the output generated by the SMDP to find simple approximately optimal policies.

The rest of this paper is organized as follows. In Section **??**, we briefly review related research. Next, in section **??**, we describe the methodology used to solve the SMDP. Furthermore, a brief introduction is given about decision tree analysis. Section **??** contains the main findings of the paper including the decision tree analysis. In Section **??**, we quantify the accuracy of the approximate policies, and study the state space size of the studied SMDPs. Finally, Section **??** concludes the paper.

II. Related Work

There has been considerable recent interest within the research community on delivery policies for adaptive video.

Many have used dynamic programming techniques, including Markov decision process (MDP) models, to optimize rate selection and video scheduling policies. A similar formulation to ours is described in [**?**], where an MDP was used to find a policy for delivery of scalable video content in an i.i.d on/off channel. The result was a "diagonal" policy that combines prefetching lower layers with backfilling upper layers. Others have considered more sophisticated scenarios and enhancements to the basic MDP formulation. For example, [**?**] develops an MDP to improve both video quality and playback smoothness, and [**?**] compares a myopic policy (that discounts the reward for video segments in the far future) and a foresighted policy.

A major issue in MDP-based techniques is that the system dynamics must be known a priori in order to solve the optimization problems, a condition that is often impractical for wireless video delivery. For this reason, we codify a set of rules for designing video delivery policies in new wireless environments (Section **??**); others have developed online learning techniques for dealing with new environments. A reinforcement learning method that learns the channel dynamics online is described in [**?**], and the method is shown to asymptotically converge to the optimal video delivery policy. Another online method, which results in a nearly optimal delivery policy for scalable video, is proposed in [**?**]. In [**?**], several techniques are proposed to reduce the computational overhead of the MDP for adaptive video delivery, including both online and offline approaches. A scheduling algorithm based on insights from an MDP solution, but requiring only partial knowledge of channel, dynamics is described in [**?**].

All of these works, however, consider unlimited buffering at the receiver. Furthermore, the optimization is performed in a very specific wireless environment. In this work, we consider for the first time the effect of content provider-enforced buffer restrictions on the receiver and examine the optimal download policies in different wireless environments.

III. Methodology

In this section, we will first give a detailed explanation of the modeling of the access constraints under study and the wireless environments in which the analysis is performed. The formulation of the SMDP is described next, followed by a brief introduction to the decision tree classifier method, which is used for deriving approximations of the SMDP policy.

A. Buffer Restriction Model

An adaptive scalable video player typically works as follows. The application has a buffer in memory or on hard disk, into which *blocks* of video identified by a combination of a *segment* index and a *layer* index are downloaded in preparation for playback. Typically the video player will begin decoding blocks for the segment with index $i+1$ as it begins playing segment i, so that segment $i+1$ will be decoded in time for playback, and will only download blocks with segment index $i+2$ or higher. We therefore consider the blocks for segments i and $i+1$ to be stored in a *playout buffer* distinct from the *download buffer*. This scenario is illustrated in Figure **??**, where the blocks to the left of the solid vertical line are in the playout buffer and the blocks to the right of the line are in the the download buffer. At each *decision epoch*, the player can choose between blocks from different layers to be retrieved into the download buffer. These blocks are called *download candidates* for that decision epoch. To keep the state space manageable, we only allow up to one download candidate per layer. That is, if the highest segment index for a block of layer l in the download buffer is i, it may download the layer l block for segment $i+1$ but not the block for segment $i+2$.

In practice, many video streaming services impose a *buffer limit* that limits how far ahead of playback a video may be retrieved. In this case, the number of download candidates may be constrained. Figure **??** shows an example in which the lower layer has already been downloaded into the first segment in the download buffer. For a client with a large buffer limit, we can identify two download candidates for the next decision instant. For a client with a buffer limit of one segment, only videos for the first segment in the buffer may be retrieved, leaving only one download candidate.

B. Wireless environment

The wireless link over which our scalable video service operates is modeled as a Markov chain with four states, with data rates of $\{0.5, 1, 2, 4\}$ Mbps respectively. We include in our study five models for the channel dynamics (i.e., the probability of transition from one state to the next):

- The **i.i.d** channel has an equal probability of moving to any state next, regardless of the current state, and models a fast varying channel
- The **low correlation** channel has a slightly higher probability of remaining in the same state or reaching a neighboring state. The variations are relatively fast in this channel.
- The **high correlation** channel has a much higher probability of remaining in the same state or reaching a neighboring state, which results in a channel with slow variations.
- The **empirical urban** channel has transition probabilities based on measurements of wireless signals in Brooklyn, NY [**?**], [**?**],
- The **empirical suburban** channel has transition probabilities based on measurements of wireless signals in Amherst, MA [**?**], [**?**].

These channel models are illustrated in Figure **??**.

C. SMDP Formulation

The video is divided into segments, each having a playback duration of one second. Each segment is encoded into one base layer and one enhancement layer of equal size. The SMDP formulation consists of a set of *states*, defining the state of the ongoing download process in any time instant. From each state, a set of *actions* is available which can be taken to move from one state to the next. The form of the actions and states is as follows.

The action is indicated by the layer number of the next download. The set of available actions in each state depends on the buffering constraint as shown in Figure (**??**)

Each state of the process is represented by the content of the download buffer, the content of the playout buffer, the time elapsed from the playback start of the current segment, and the instantaneous channel quality. The download buffer is represented by a vector whose entries indicate how many future segments have been retrieved for each of the layers. The playout buffer has a length equal to three segments and is represented by a vector indicating the number of layers of video in each of these three buffer position.

Time is assumed to be slotted and the duration of a time slot is equal to the fastest action being performed, which is downloading a segment in the best state of the channel. All other actions take approximately integer multiples of a timeslot.

The perceived video quality is modeled as a sublinear function of the video rate described in [**?**]. At each state, a reward $r_{s,a}$ is assigned for every possible action as follows:

$$r_{s,a} = \sum_{t=1}^{t_{s,a}} e^{-\alpha\left(\frac{R_t}{R_{max}}\right)^{-\beta}+\alpha}, \tag{1}$$

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 $t_{s,a}$), R_t is the rate of the video being played back at instant t and R_{max} is the maximum video rate. The constants α and β are video specific and are set to 0.16 and 0.66, respectively, as suggested in [**?**]. suggested in To solve the SMDP, we apply the value iteration method [**?**].

The outcome of the SMDP is the mapping from each state to the respective optimal action in that particular state. However, there are states from which the same value is obtained for base and enhancement layer download. In other words, it does not matter which action is taken. We refer to these actions as indifferent actions. Hence, there are three possible actions in each state in total.

D. Derivation of approximate policies

The SMDP policy we computed, is essentially a classifier, which takes as its input a 5-dimensional state, and classifies it into one of three action groups ("classes"). To gain insight into the output of the SMDP policy, we use a decision tree classifier [**?**] to develop a simplified set of rules for selecting the next block to retrieve.

We use the `party` [**?**] package in *R* to construct our decision tree. The tree is trained on a set of data points (in this case, the states from the SMDP) and the action as determined by the SMDP. It distills this complex multidimensional dataset into a set of rules that represents the SMDP as accurately as possible.

Each node of the tree partitions the input dataset by splitting it on one of the state parameters (or some transformation of them). A sample tree is illustrated in Figure **??**. We allow the tree to split on any of the following parameters: the channel state (c), the duration of the unplayed video in the playback buffer (p), the number of blocks in the download buffer for each of three layers (n_0, n_1, n_2), and the difference in the number of blocks in the download buffer between any two consecutive layers ($d_{0,1}, d_{1,2}$). The `party` library recursively tries different split parameters and different split values to find a good split (one that effectively reduces the entropy of the output). We limit the depth of the tree to 3 so that the rule-based policy is not overly complex.

The result of the decision tree not only gives us a simple policy that approximates the SMDP policy, but also gives us some measure of the relative importance of different state parameters in determining the best action. For example, parameters that appear near the root of the tree affect more cases, and can be considered more important than those that appear at the bottom of the tree. Similarly, split conditions that appear often in the tree regardless of channel dynamics can be considered more "universal" than those that only appear under certain conditions.

IV. Discussion

For each of the presented channel models, a SMDP is set up for different buffer limit values ranging from 5s to 100s with a discount factor of 0.95. The decision tree algorithm with a tree depth of 3 is then applied to the outcomes of each scenario.

The results of the decision tree indicate that the generic rule that governs over all scenarios is that the frontier of the base layer buffer never falls behind that of the enhancement layer buffer. In other words, the diagonal policy introduced in [**?**] is also observed here. The slope of this diagonal, which is the difference between the number of base and enhancement layer segments in the buffer, is a function of both the buffer limit and the pace of the channel variations as will be described next.

The tree outputs indicate that in all scenarios, the tree root, which carries the most important attribute for decision making, is the difference between the number of base and enhancement layer segments in the download buffer. Figure **??** compares the information gain of the layer difference with that of the instantaneous channel state. The information gain of an attribute is defined as the average reduction in the entropy of the dataset, if it was split by that attribute (Reference maybe?). The higher the information gain, the more impact that attribute has on the overall policy. It can be seen that for all cases, the layer difference is a more determining factor than the channel state. It is also worth noting that the significance of the channel state decreases with buffer limit.

For each scenario, there is a threshold for the layer difference, below which the algorithm tends to request base layer segments and above which it tends to request enhancement layer segments. In figure **??**, this threshold is depicted for different channels with respect to the buffer limit. A higher threshold indicates a more conservative download policy because it means that the receiver prefers a safer base layer margin to reduce the danger of starving the buffer. On the other hand, a lower threshold represents a greedy behaviour in which the receiver starts downloading enhancement layers at an earlier stage to improve quality.

As it can be seen, fast varying channels have more greedy download policies than slow varying channels. The reason for this is that in a slow varying channel, a deep fade state is hard to recover from due to the high temporal correlation of the channel quality. In other words, if the capacity drops low, it will stay low for a significant amount of time. Consequently, the optimal strategy in these environments is to have a large base layer margin in order to prevent buffer starvation.

However, there are exceptions to this thresholding policy at which the optimal strategy is to take the indifferent action. A user would choose to act indifferently, if at one hand there is enough base layer segments in the buffer to avoid potential buffer starvation, and on the other hand, there are enough enhancement layer segments to avoid a loss in video quality. The indifferent choice of action can be considered as a loose policy, which is taken in safe conditions. Opposingly, the strict actions of choosing a specific layer to download is an indicator of either avoiding buffer starvation or improving quality. Figure **??** shows the cases in which the indifferent action is optimal for different channel variations at three buffer limits of 20, 60 and 100 seconds.

It should be noted that in Figure **??** and **??**, the actual values of the thresholds are also a function of the discount factor. However, it is observed that in all investigated scenarios, the relative differences in the policies for different environments follows the same pattern.

According to this plot, once the number of both base and enhancement layer segments in the download buffer exceeds a certain threshold, the optimal action becomes indifferent of the layer index. As it can be observed also in this figure, the threshold is higher for channels with slower variations. Similar to the previous case, it can be argued that the conservative strict policy of dowloading a specific layer is favored in slowly varying channels for a longer period. On the other hand, in fast varying channel, the loose indifferent policy kicks in sooner.

The above analysis holds true for buffer limits beyond 20 seconds. In these scenarios, the instantaneous channel state is a very insignificant attribute in the decision policy. Precisely, for these buffer limit, the channel state never even appears in a tree of depth 3. However, this changes in small buffer limit scenarios. It turns out that if the buffer limit is small, although the generic rule of prefetching and backfilling still holds, the instantaneous channel state becomes a determining factor in the decision making.

Furthermore, as shown in Figure **??**, the effect of instantaneous channel state is more visible in slow varying environments. A similar explanation to the large buffer scenario can be used here as well. The slower the variations, the longer the channel remains in a particular state. If also the amount of buffered data is very limited, the danger of emptying the buffer in a long lasting bad channel is high. Therefore, in these environments, the policy is very conservative in bad states and turns greedy in good states. On the other hand, if the channel capacity varies fast, the uncertainty about the future channel state is high, which makes the current state of the channel non-suitable for decision making.

V. Remarks

The presented results in this paper are based on a decision tree of depth three. Decision trees with higher depth have a higher accuracy but they are also more complicated to evaluate. Figure **??** shows the accuracy of the decision trees in each scenario. The accuracy is calculated based on the relative number of correctly classified versus the total number of states in each case. According to this figure, the accuracy will never be less than 85%, which we argue to be acceptable to derive the approximate policies.

Figure **??** shows the number of states generated by the SMDP which is the same for all channel types being used. Large state spaces are a limitation to the settings of the SMDP and can increase drastically with the buffer limit, number of enhancement layers and the set of available data rates. The higher the number of states, the more complex and time consuming the value iteration and classification.

VI. Conclusion

In this paper, the effect of wireless channel characteristics on optimal SVC download policies is investigated under varying buffer constraints using the decision tree method. The results indicate that the general policy is to prefetch base layers ahead and backfill enhancement layers afterwards. Furthermore, the margin of base and enhancement layer difference is larger for slow varying channels, which implies a more conservative policy in these cases. The instantaneous channel is only significant in small buffer scenarios and mostly affects the slowly varying environments.

In this work, only the single user scenario is considered and the more general case of a multiuser scenario in a networking environment where users are competing for resources, is left for future work.