Signal Processing: *Image Communication* 39 (2015) 305–315

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

Dynamic rate adaptation for adaptive video streaming in wireless networks ☆

Siyuan Xiang [a], Min Xing [a], Lin Cai [a], Jianping Pan [b]

[a] *Department of Electrical & Computer Engineering, University of Victoria, Victoria, BC, Canada*
[b] *Department of Computer Science, University of Victoria, Victoria, BC, Canada*

ARTICLE INFO

Article history:
Received 8 April 2015
Received in revised form
26 June 2015
Accepted 14 August 2015
Available online 28 September 2015

Keywords:
DASH
Wireless multimedia

ABSTRACT

In this paper, we investigate the streaming strategy for dynamic adaptive streaming over HTTP (DASH). Specifically, we focus on the rate adaptation algorithm for streaming scalable video (H.264/SVC) in wireless networks. We model the rate adaptation problem as a Markov Decision Process (MDP), aiming to find an optimal streaming strategy in terms of user-perceived quality of services (QoS) such as playback interruption, average playback quality and playback smoothness. We then obtain the optimal MDP solution using dynamic programming. However, the optimal solution requires the knowledge of the available bandwidth statistics and has a large number of states, which makes it difficult to obtain the optimal solution in real time. Therefore, we further propose an online algorithm which integrates the learning and planning process. The proposed online algorithm collects bandwidth statistics and makes streaming decisions in real time. A reward parameter has been defined in our proposed streaming strategy, which can be adjusted to make a good trade-off between the average playback quality and playback smoothness. We also use a simple testbed to validate our proposed algorithm. Experimental results show the feasibility of the proposed algorithm and its advantage over the existing work.

1. Introduction

Current statistics show that video applications account for the highest percentage of the traffic mix in the Internet. Cisco forecasts that, by 2018, the sum of all forms of video, including TV, video on demand, Internet video, and peer-to-peer (P2P) video, will account for 79% of the global consumer network traffic, while mobile video will account for more than 75% of the total mobile network traffic [2].

The increasingly popular video websites such as YouTube and Vimeo will be the major providers of mobile videos. Progressive download is currently the dominant video delivery techniques of these video websites. It has several advantages over the traditional streaming techniques using RTP/UDP. First, it is simple to deploy. At the server side, any web server can host videos and serve as a streaming server; at the client side, the user only needs a flash player or web browser supporting HTML5 for video playback. Second, the HTTP/TCP protocols used in progressive download are more firewall and network address translation (NAT) friendly, and the congestion control mechanism in TCP simplifies the design of the application layer. Third, for progressive download, a server can store several versions of a video to meet the requirements of heterogeneous users, so a user can select the right version of the video according to the device decoding capability, display size and available network bandwidth.

☆ Preliminary results of this paper have been presented at ACM MMSys'12 [1]. New technical contributions, including the design of the online algorithm, the video segment storage structure, the SVC video player implementation and the experiments using the wireless testbed, are presented in Sections 4 and 5, respectively.

http://dx.doi.org/10.1016/j.image.2015.08.007
0923-5965/

However, selecting the appropriate version of a video to match the available bandwidth may not be easy for users and their decisions might be error-prone. In addition, with progressive download, the client always downloads as much video data as possible. Plissonneau and Biersack [3] report that only half of the videos are fully downloaded and this number drops dramatically when the users are not satisfied with the video quality. It is likely that when a user turns off the video player or switches to another video, a large amount of un-watched video has been buffered unnecessarily, which wastes the resources of both the network and the end-systems. It is particularly undesirable for mobile devices with limited energy supply.

Dynamic adaptive streaming over HTTP (DASH) [4] is a promising technique to overcome the aforementioned disadvantages of progressive download. Videos encoded in different versions are chopped into small segments. After the client receives one segment, it has a chance to decide which version of the video to request for the next segment, based on the current network condition. Thus, rate adaptation can be performed at the client side naturally and flexibly. Also, the client has a chance to control the client-side queue length to avoid streaming buffer overflow, e.g., when the download rate is much higher than the playback rate.

Currently, commercial adaptive streaming products such as Microsoft Smooth Streaming and Apple Live Streaming support single-layer H.264 advanced video coding (AVC) encoded videos. Multiple versions of a video with different resolution, frame rate and quality are obtained by encoding the source video multiple times with different configurations, and the different versions of the video are completely independent of each other. Thus, not only more server storage space is needed, but also the web caching hit-ratio is reduced.

Recently, scalable video coding (H.264/SVC) has been introduced to the DASH framework to improve the system performance [5]. With SVC, a video is encoded once only, and it can be decoded many times with different resolution, frame rate and quality. However, how to improve the rate adaptation algorithm to provide users with a satisfactory quality of services (QoS) is still a challenging and open question. The problem is even more challenging when a user uses a handheld device via a wireless access link for video streaming, as the handheld devices typically have limited energy supply and computation capacity, and the wireless links are highly dynamic due to the time-varying fading, shadowing, interference and hand-off, all of which motivated our work.

In this paper, we design the rate adaptation algorithm for streaming scalable video over HTTP in wireless networks. The main contributions of this paper are threefold. First, we formulate the rate adaptation problem as a finite Markov Decision Process (MDP), aiming to find an optimal streaming strategy in terms of user-perceived QoS such as playback interruption, average playback quality and playback smoothness. We obtain the optimal streaming strategy by dynamic programming under the reinforcement learning framework [6]. A reward parameter is defined in our proposed strategy, which can be adjusted to make a trade-off between average playback quality and smoothness. Second, since the optimal solution requires the knowledge of available bandwidth statistics and has a high computational complexity, which makes it difficult to obtain the optimal solution in real time, we propose an online algorithm which integrates the learning and planning process, i.e., the proposed algorithm collects bandwidth statistics and makes streaming decisions in real time. Third, we have prototyped a scalable video streaming framework including the server-side video pre-processing and client-side SVC video playing back. A real sample video encoded in SVC is used to evaluate the proposed streaming strategies and compare them with the existing work using both wireless testbed experiments and simulations. The experimental and simulation results show the advantage of the proposed algorithms in practical settings.

The rest of the paper is organized as follows. Section 2 summarizes the related work. Section 3 formulates the optimal streaming problem as an MDP. Section 4 presents the proposed optimal streaming policy and the online algorithm. The evaluation framework, testbed configurations and experimental results are described and given in Section 5, followed by concluding remarks and further research issues in Section 6.

2. Background and related work

Different from the application-layer multicasting [7], in a DASH system, rate adaptation is conducted at the client side, which is also called pull-based rate adaptation [8]. At the server side, a source video is encoded into different versions with different resolution, frame rate and quality. For each version, the video is divided into small segments. A web server can host these segments and send them to the clients upon HTTP requests. At the client side, after a user clicks the play button, the streaming starts. The video player first obtains the general information of the video, such as the number of different versions and the corresponding resolution, frame rate and quality of each version. Then, the video player will decide the right version according to its own display size, decoding capability and network condition. Usually, the playback does not start until a sufficient number of segments are received. After the client receives a segment completely, the rate adaptation algorithm will decide which version to request for the next segment based on the current network condition and the client-side state such as the number of buffered segments. In this way, the workload of the server is reduced dramatically. Fig. 1 shows the general workflow of the video player in a DASH-like system.

There are extensive research efforts on the adaptive video streaming over HTTP [4,9,10,11]. Stockhammer [4] introduced the 3GPP specification of the dynamic adaptive streaming over HTTP, which describes the framework of the adaptive streaming system. In [10], the commercial adaptive streaming products including Microsoft Smooth Streaming and Netflix player and the open source media framework (OSMF) player were evaluated and compared. The results show that the performance of these products still needs to be improved substantially.