A QoS-Adaptive Framework for Screen Sharing Over Internet

Duc V. Nguyen¹, Huyen T. T. Tran¹, Pham Ngoc Nam² and Truong Cong Thang¹
¹The University of Aizu, Japan
²Hanoi University of Science and Technology, Vietnam
Email: {m5182102, m5192110, thang}@u-aizu.ac.jp, nam.phamngoc@hust.vn

Abstract—In this paper, we present a QoS-adaptive framework for screen sharing over heterogeneous networks. Basic requirements and important issues of a screen sharing system are described. For determining optimal video coding parameters, adaptation trajectory is used as the hints in making decisions. Further, experiments are carried out to investigate the trade-off between video bitrate and the buffering delay in time-varying networks.

Keywords—Adaptive Streaming, Screen Content.

I. INTRODUCTION

Thanks to the availability of many smart devices, displays, and broadband connections, screen casting/sharing has become an important functionality for user devices [1]. In such an application, the screen content of a device is encoded as images/videos and then delivered to other devices. Important applications based on screen content are screen casting, e-learning, conferencing, wireless display, etc. [1], [2]. A key characteristic of screen content is the presence of graphics and texts, which are not effectively coded by previous video coding formats [3]. Meanwhile, the important requirements for any screen content application are effective content coding, seamless sessions, and low delay. Early studies have presented special coding solutions for screen content, where different spatial and temporal characteristics are exploited [3]. Also, transport mechanisms were investigated for screen sharing to a single user as well as multiple users [3], [4].

To meet the market demand for screen casting/sharing, a new video coding standard, which is called Screen Content Coding (SCC) and essentially an extension of the recent High Efficiency Video Coding (HEVC) standard [5], is going to be issued in 2016 [6]. This standard includes various coding tools to handle screen content. Moreover, as it is backward compatible to HEVC version 1 and HEVC range extension (HEVC-RExt) [6], SCC has all advantages of these latest video standards in coding camera-captured content.

Recently, some studies have evaluated the feasibility of new video coding formats and video delivery solutions over IP networks [7][8]. However, existing evaluations are either for non-adaptive delivery [7] or for camera-captured videos with high delay [8]. They are not appropriate for SCC, which has different content characteristics and QoS requirements. Some other studies have proposed systems for low-delay screen sharing and cloud gaming [9][10]. However, their systems are based on UDP [9] that is not well supported today. Although the system proposed in [10] can run over both UDP and TCP, the evaluation has only considered the network with very stable bandwidth.

In this paper, we propose a QoS-adaptive framework for screen sharing over the Internet. For determining optimal video coding parameters, adaptation trajectory is used as the hints in making decisions. The proposed system is able to adapt the video bitrate to available throughput and select the optimal video coding parameters for the best possible video quality. To the best of our knowledge, this is the first study that deals with streaming of screen content over the Internet.

The remainder of the paper is organized as follows. The architecture of the proposed system is given in Section II. Section III presents the experiment results. The paper is concluded in Section IV.

II. PROPOSED SYSTEM

A. System architecture

Fig. 1 shows the general architecture of our proposed system where a host B is sharing its screen with a host A. Communication between the two hosts is based on TCP. The host B sends video data to the host A in frame basis. During a streaming session, the host A periodically sends feedback containing information about available throughput to host B.

The host A consists of three components as follows. The Receiver is responsible for receiving video data from the host B. The task of the Monitor is to estimate the available network...
throughput. The Feedback Reporting periodically sends the estimated throughput computed by the Monitor to the host B.

The host B also consists of three components as follows. The Decision Engine decides the optimal encoding parameters given an estimated throughput. The task of the Encoder is to compress the raw screen images based on the parameters decided by the Decision Engine. To satisfy the low delay requirement, the Encoder uses the low-delay B coding structure with Group of Picture (GOP) size of 4 as shown in Fig. 2. With this structure, the Encoder can start encoding a frame as soon as it becomes available. The Sender reads the encoded video from the Encoder’s buffer and sends to the host A in frame basis.

Note that, the Decision Engine can take other inputs such as the receiver’s buffer level. In our proposed system, the inputs are made available to the Decision Engine by having the receiver periodically sends feedback to the sender. Since it takes at least one half of the network delay for the feedback to arrive at the sender, the information used by the Decision Engine is delayed. Using the delayed information will cause inappropriate decision that may lead to negative impacts to the receiver such as buffer underflow. To eliminate the impact of the delay caused by feedback delay, a possible solution is to estimate the available throughput and the receiver’s buffer level at the sender. The sender can estimate the available throughput and the receiver’s buffer level by monitoring the amount of data it sends to the receiver.

![Fig. 2. Low-delay B coding structure with GOPsize=4.](image)

**B. Bandwidth Estimation by the Monitor**

The Monitor estimates the available throughput as follows. A throughput sample is computed every frame by dividing the frame size to the time required to receive that frame. Every S frames, an estimated throughput is computed as the average throughput of the last S received video frames.

**C. Decision Engine**

Given an estimated throughput, the task of the Decision Engine is to decide a video bitrate and the corresponding coding parameters for the best possible video quality while meeting the requirements on delay and buffer stability.

Assume that the requesting content is provided at M versions. Version i has bitrate $R_i (1 \leq i \leq M)$. The video versions are generated by controlling three coding parameters, namely Quantization Parameter ($QP$), Frame Rate ($FR$), and Resolution ($RE$). Given an estimated throughput $T^*$, the adaptation problem can be stated as follows.

Find a version $j$ and the coding parameters ($QP, FR, RE$) so as to maximize the resulting video quality $Q$ and satisfy the constraints.

$$R_i \leq T^*.$$  \hspace{1cm} (1)

In this paper, we adopt the instant throughput (ITB) method [8] to decide the bitrate $R_i$. Specifically, given an estimated throughput $T^*$, the bitrate $R_i$ is computed by,

$$R_i = \max \{R_j | R_j \leq (1 - \alpha) \times T^*, 1 \leq i \leq M\},$$  \hspace{1cm} (2)

where $\alpha (0 \leq \alpha \leq 0.5)$ is a safety margin for compensating the mismatch between the estimated and the real throughput.

To determine the optimal coding parameters given a bitrate value, we have conducted subjective tests to identify the adaptation characteristics of different SCC videos. At a given bitrate, each original video is encoded into multiple versions of equivalent bitrates. Each version has different frame rate, resolution, and QP. The generated versions are evaluated by a number of viewers. The version that has the highest average score is selected as the best version. The set of best versions at different bitrate values constitutes an adaptation trajectory of the original video [12].

Fig. 3 shows the adaptation trajectory of the SlideEditing video [11]. The trajectory is composed of segments $O_i P_i$ where the point $O_i$ corresponds to the best-quality version. On each segment $O_i P_i$, the frame rate and the resolution are constant, and the video is adapted by increasing the QP value (i.e., decreasing quality level). Point $P_i$ and point $O_{i+1}$ have the same bitrate. It can be seen that when the bitrate is reduced, the user wants to keep the original frame rate and resolution, however, only increase the QP, i.e., segment $O_i P_i$. As the bitrate is further reduced, the user wants to reduce the frame rate (i.e., 1fps), and keep the original resolution, i.e., segment $O_i P_{i+1}$. This indicates that one should adapt this type of video content in quality dimension first and then temporal dimension without reducing video resolution. To support online adaptation, we assume that the adaptation trajectory is stored on the sending host (i.e., the host B) in advance for each type of videos.

![Fig. 3. The adaptation trajectory of the SlideEditing video.](image)

**III. EXPERIMENTS AND RESULTS**

In this Section, we will investigate performance of the proposed system in two aspects: Optimal selection of coding parameters, and bitrate adaptation in fluctuating bandwidth conditions.
Table I shows the optimal QP values and frame rates at different bitrate values for the SlideEditing video. If the bitrate is equal to the bitrate of a point in the trajectory, the optimal QP and frame rate can be decided immediately. Otherwise, those values can be determined by interpolating the neighboring points.

**TABLE I.** **OPTIMAL QP VALUES AND FRAME RATES AT DIFFERENT BITRATE VALUES FOR THE SLIDEEDITING VIDEO.**

<table>
<thead>
<tr>
<th>α</th>
<th>QP</th>
<th>Frame rate (fps)</th>
<th>MOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>22</td>
<td>30</td>
<td>4.9</td>
</tr>
<tr>
<td>0.2</td>
<td>26</td>
<td>30</td>
<td>4.9</td>
</tr>
<tr>
<td>0.3</td>
<td>30</td>
<td>30</td>
<td>4.6</td>
</tr>
<tr>
<td>0.4</td>
<td>34</td>
<td>30</td>
<td>4.1</td>
</tr>
<tr>
<td>0.5</td>
<td>42</td>
<td>30</td>
<td>3.1</td>
</tr>
<tr>
<td>0.6</td>
<td>42</td>
<td>1</td>
<td>2.9</td>
</tr>
<tr>
<td>0.7</td>
<td>27</td>
<td>1</td>
<td>2.6</td>
</tr>
<tr>
<td>0.8</td>
<td>34</td>
<td>1</td>
<td>2.3</td>
</tr>
</tbody>
</table>

Next, we will investigate the proposed system in fluctuating network conditions using a real bandwidth trace as shown in Fig. 4. Video content has frame rate of 30fps. There are 9 versions with bitrate from 400kbps to 3200kbps, and step-size of 400kbps. The client buffer size and the number of throughput samples are set to 960ms and 1, respectively. Network round-trip time delay is set to 10ms. We use two performance metrics: average bitrate, and maximum buffer variation.

Table II shows the average bitrate and the maximum buffer (level) variation at different safety margin values. Note that the smaller the maximum buffer variation is, the smaller the buffer size and so the initial buffering delay can be. It can be seen that, the higher the value of α is, the lower the average bitrate and the maximum buffer variation become. When α = 0.1, the average bitrate is the highest but it also results in the highest buffer variation of 859.38ms. When α = 0.5, the buffer variation could be below 100ms but the average bitrate is reduced by a factor of 1.7. Therefore, the safety margin α allows us trading off between the video quality (i.e., average bitrate) and buffering delay (i.e., buffer variation). It is obvious that the appropriate value of α depends on a given application’s requirements. Our future work will further study the trade-off of different factors in screen sharing systems.

**TABLE II.** **AVERAGE BITRATE AND RECEIVER’S MAXIMUM BUFFER VARIATION WITH DIFFERENT SAFETY MARGIN VALUES**

<table>
<thead>
<tr>
<th>α</th>
<th>Avg. Bitrate (kbps)</th>
<th>Max. Buffer Variation (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1726.00</td>
<td>859.38</td>
</tr>
<tr>
<td>0.2</td>
<td>1634.19</td>
<td>456.78</td>
</tr>
<tr>
<td>0.3</td>
<td>1487.54</td>
<td>229.60</td>
</tr>
<tr>
<td>0.4</td>
<td>1268.13</td>
<td>148.89</td>
</tr>
<tr>
<td>0.5</td>
<td>1020.49</td>
<td>62.18</td>
</tr>
</tbody>
</table>

In this paper, we have proposed and implemented a QoS-adaptive framework for screen sharing over the Internet. For determining optimal video coding parameters, adaptation trajectory was used as the hints in making decisions. Experiments were carried out to investigate the trade-off between video bitrate and the buffering delay in time-varying bandwidth conditions. In the future work, we will further study the trade-off of different factors in screen sharing systems.

**REFERENCES**


