Playout Buffer based Rate Adaptation for Scalable Video Streaming over the Internet

Youngwook Kang, Young H. Jung, and Yoonsik Choe

School of Electrical and Electronic Engineering
Yonsei University
Seoul, Korea
E-mail: {nanwook, crosscom, yschoe}@yonsei.ac.kr

ABSTRACT

The use of scalable video coding scheme has been regarded as a promising solution for guaranteeing the quality of service of the video streaming over the Internet because it is a capable coding scheme to perform quality adaptation depending on network conditions. In this paper, we use a streaming model that transmits base layer using TCP and enhancement layers using DCCP, which try to provide transmission reliability of the BL and TCP friendliness. Unlike pervious works, the proposed algorithm performs rate adaptation based on playout buffer status. The PoB status of the client is sent back periodically to the server and serves as a network congestion indicator. Experimental results show that our scheme improves streaming quality comparing with pervious scheme in the case of not only constant/dynamic background flows but also VBR-encoded video sequence.

Keywords: scalable video streaming, rate adaptation, quality of service, TCP, DCCP

1. INTRODUCTION

With the fast development of video coding schemes and the Internet infra-structure, video streaming service over the Internet has become one of the most popular applications among network based services.

In generally, the encoded video data is very sensitive to packet loss and delay variation during the data transmission because most of the video coding schemes include the variable length coding (VLC) and motion vector prediction. However, the video streaming service in the current Internet is transmitted by the only best-effort manner. Therefore, it is inevitable to suffer from bandwidth fluctuation, packet loss and delay variation. In this situation, guaranteeing the quality of video streaming despite such packet loss and delay variation has become a crucial issue.

The use of scalable video coding scheme has been regarded as a promising solution for guaranteeing the quality of service (QoS) of the video streaming over the Internet because it is a capable coding scheme to perform quality adaptation depending on network conditions [1]. SVC-encoded bitstream consists of a base layer (BL) and several enhancement layers (EL). The BL provides not only a basic quality of video streaming but also improved qualities when combined with EL streams. Thus, the BL should be handled more important than other layers. Since this unequal importance among layers naturally leads to use of unequal error protection (UEP) policy, UEP has been widely adopted in SVC-encoded video streaming scheme [2][3]. Although the use of SVC with UEP can present a better performance in video streaming over error prone environment, it needs additional redundancy parity packets and information about error bound [4]. Moreover, when the FEC is used with UDP, the UDP streams with large bandwidth for providing high-quality video streaming service may lead to congestion collapse in the limited network [5].

Gorkemli et al. investigated the method for SVC-encoded video streaming both using TCP and DCCP [6]. They transmitted BL using TCP and transmitted ELs using DCCP, which tried to ensure not only transmission reliability of the BL without UEP but also TCP-friendliness. However, in this streaming model, the TCP and the DCCP flows competed with each other and other flows within available bandwidth. As a result, re-transmission of TCP flow yielded re-buffering event and packet loss of DCCP flow yielded distortion. To minimize re-buffering event and maximize streaming quality, [6] performed rate adaptation by using congestion controlled DCCP rate. Although it could perform suitable rate adaptation over constant background flows, in the case of dynamic background flows or VBR-encoded video sequence, it could not perform proper rate adaptation according to congestion occurrence and alleviation. As a result, large quality degradation was presented by late packets of EL.

In this paper, we propose a rate adaptation algorithm that can apply to the situation of not only constant/dynamic background flows but also VBR-encoded video streaming over TCP/DCCP. Unlike the previous Gorkemli's algorithm that performs rate adaptation only depending on the DCCP rate, the proposed algorithm is based on playout buffer (PoB) status. The proposed algorithm is composed of PoB estimator, congestion level determinator (CLD) and GoP-based average encoding rate calculator. The rate adaptation module performs rate adaptation according to congestion level which is determined by the PoB-estimator and the CLD. The rest of the paper is organized as follows. Section 2 presents the SVC coding features and previous rate adaptation scheme. In section 3, we introduce the proposed rate adaptation algorithm including the PoB estimator, CLD and GoP-based average encoding rate calculator. The simulation results and performance evaluation are described in section 4. Conclusions are given in section 5.

2. BACKGROUND

2.1 Scalable Video Coding (SVC)

The SVC supports three types of scalability that are spatial, temporal and quality scalability. For supporting spatial scalability, each spatial layer is encoded by using motion, texture and residual signal resulting from encoding process of lower layer. Therefore, when each layer is referred, motion and texture information is changed by the ratio of resolution between layers. To improve coding efficiency through adequate reference between layers, inter-layer prediction schemes were incorporated. For supporting temporal scalability, the hierarchical B-picture was incorporated. Since the memory management control operation (MMCO) commands in H.264/AVC can control the decoded picture buffer (DPB), reference frames can be optionally selected by using the reference picture list re-ordering (RPLR) commands. In this way, general B-picture was defined for using B-picture as reference image and was basic concept of the hierarchical B-picture structure. Quality scalability can be considered as a special case of the spatial scalability with identical picture sizes for base and enhancement layer [1]. In the SVC, two different possibilities exist for providing quality scalability. The first one is referred to as coarse-grain scalability (CGS) and the second one is referred to as median-grain scalability (MGS). The CGS can be achieved by re-quantizing the residual texture signal of enhancement layer with smaller quantization step size relative to that used for preceding CGS layer. The MGS was proposed to increase the flexibility of rate adaptation and to improve the coding efficiency by providing a variety of bit rates. The MGS can choose to partition the transform coefficients of a layer into up to 16 MGS layers. In this way, fine granularity can be achieved.

Although coding efficiency of SVC is a little lower than single layer video coding, the SVC can perform more flexible quality adaptation according to unpredictable network condition. In this paper, we use the quality scalability for simulation convenience.

2.1 Rate adaptation algorithm for streaming using TCP/DCCP

In [6], Gorkemli et al. proposed a streaming model using both TCP and DCCP. Fig.1 illustrates Gorkemli's system architecture. The BL is transmitted by TCP and the ELs are transmitted by DCCP, through which minimum quality of video streaming can be ensured by reliable BL transmission and congestion collapse can be avoided by TCP-friendly DCCP flow. However, TCP and DCCP flows compete with each other and other flows because link capacity of network is limited. For example, the DCCP flow gets more bandwidth share, the TCP flow gets less bandwidth share and vice versa. As a result, this leads to increasing transmission delay or quality degradation.

To minimize the rebuffering event and to maximize the quality of video streaming, [6] performed some rate adaptation schemes : slow-increasing/fast-decreasing, fast-increasing / fast-decreasing and slow-increasing / slow-decreasing sending layers. As a result, the best compromise was achieved when the sending layers were changed slowly. This scheme is depicted in Fig.2. The client sends back periodically to the server information of current receiving DCCP rate that means congestion controlled transmission rate. The server increases and decreases sending layers of EL according to the feedback information.

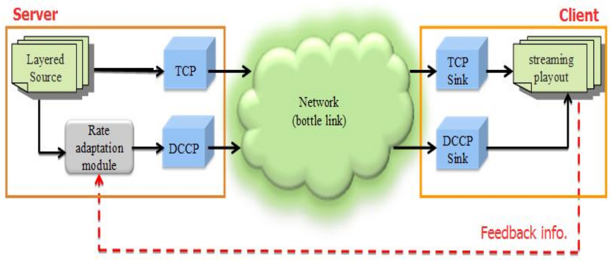


Fig.1 Streaming system architecture over TCP/DCCP

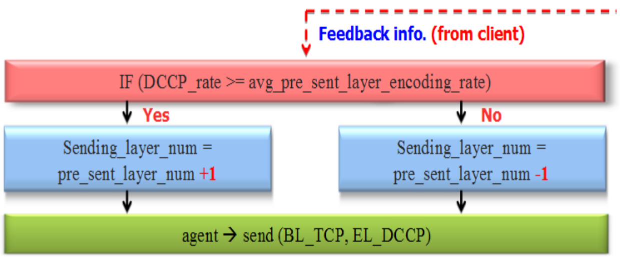


Fig.2 Gorkemli's rate adaptation algorithm

In the video streaming model using TCP/DCCP, it is important to perform rate adaptation suited to congestion occurrence or alleviation because BL and EL are transmitted by each channel. If the rate adaptation process is failure and the server sends more EL data than allowed sending rate, BL and EL packets have more time gap than expected one between sending time and receiving time. This time gap leads to large quality degradation by the packets of EL. In the case of constant background flows that means fixed cross traffic flows, we could confirm the Gorkemli's rate adaptation scheme could select the proper sending layers and improve streaming quality. However, in the case of dynamic background flows or VBR-encoded video sequence, it could not perform proper rate adaptation suited to congestion occurrence and alleviation. As a result, large quality degradation was presented by late packets.

3. PROPOSED CONTENT-AWARE AND TCP-FRIENDLY RATE ADAPTATION

3.1 Playout Buffer (PoB) Estimator

To guarantee the video streaming quality in the case of not only constant / dynamic background flows but also VBR-encoded video, we add several modules to previous Gorkemli's streaming architecture for detecting network congestion and preventing the abrupt change of sending rate. In Fig.3, the blurred hexahedron boxes represent additional modules : PoB estimator, CLD and GoP-based average encoding rate calculator. The rate adaptation module determines the sending layers by using the information from each module.

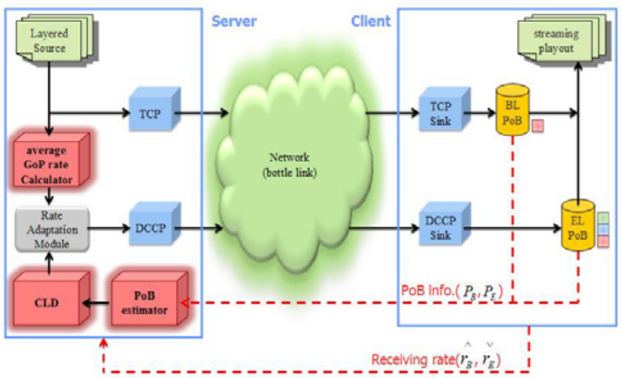


Fig.3 Proposed streaming system architecture

The PoB estimator predicts PoB status of next time slot. Each time slot is defined as feedback period and is denoted as Δt. The estimation of PoB is performed by using (1) and (2) as follows:

$$P_B(t+\Delta t) = P_B(t) + \int_t^{t+\Delta t} \hat{r}_B(t)dt - \mu_B \cdot \Delta t \qquad (1)$$

$$P_E(t+\Delta t) = P_E(t) + \int_t^{t+\Delta t} \check{r}_E(t)dt - \mu_{E\_\hat{n}} \cdot \Delta t \qquad (2)$$

where $P_B(t)$ and $P_E(t)$ represent the current PoB status of BL and EL which are feedback information from the client. The value $\hat{r}_B(t)$ and $\check{r}_E(t)$ mean the average receiving rate and the average effective receiving rate respectively. The effective receiving rate is calculated by considering only playable packets that except unusable packets such as late packets and packets of lossy frame. The μ_B and μ_E represent the average encoding rate of BL and EL which are known to the server. To calculate (2), the server needs additional information about how many EL layers are being played at the client. Therefore, we define $\hat{n}$ that represents average played layers during the time slot. The client sent $\hat{n}$ to the server as feedback information. The estimated PoB information is sent to the CLD and is used as parameter to detect network congestion. The simulation results show estimated PoB status is similar to real PoB status.

3.2 Congestion Level Determinator (CLD)

If network congestion is occurred, a little amount of the video data can be transmitted and may lead buffer starvation. Therefore, the PoB occupancy can be used as parameter for detecting the network congestion. The CLD determines congestion level according to the PoB information and threshold values. There are three scaling threshold values such as P_{B_th}, $P_{E\_low\_th}$ and $P_{E\_up\_th}$. The P_{B_th} represents the lower threshold of BL PoB. The $P_{E\_low\_th}$ and $P_{E\_up\_th}$ represent the lower and upper threshold of EL PoB respectively. These threshold values were determined by a number of simulations. In our algorithm, the lower thresholds are defined as 65% of initial buffering time and upper threshold is defined as 100% of initial buffering time. We suppose that the server sends maximum EL layers during initial buffering.

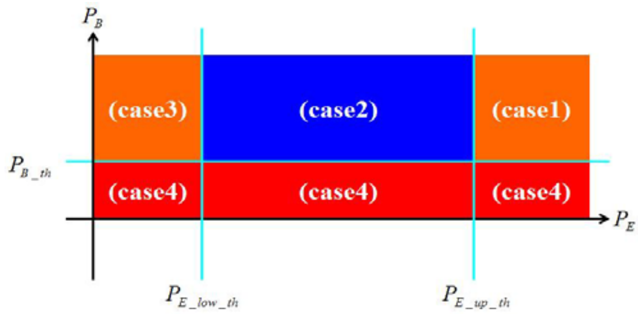


Fig.4 Congestion level determinator (CLD)

The Fig.4 illustrates the CLD. There are four congestion cases which are divided by three threshold values. The rate adaptation module applies the different rate adaptation process to each case.

3.3 GoP-based Average Encoding Rate Calculator for VBR-encoded Sequence

The VBR-encoded video sequence typically shows very strong burstness in bitrates. Therefore, its bandwidth requirement is highly variable. If the server continuously sends same EL layers in the part of burst increasing encoding rate, the sending rate is shortly increased, and it may lead network congestion. To solve this problem, the proposed algorithm calculates average encoding rate in the unit of GoP and re-calculates this at the time that every frame is sent. The rate adaptation module uses this information for preventing the burst change of sending rate.

The GoP-based average encoding rate is also used for restricting the number of maximum sending layers in congestion free condition. The proposed algorithm sets up the upper bound of sending rate according to GoP-based average encoding rate. This constraint prevents the congestion occurrence by intrinsic rate variation of VBR-encoded video sequence.

3.4 Rate Adaptation Module

Fig.5 depicts the proposed rate adaptation algorithm. It performs rate adaptation according to four congestion level such as Case1 ~ Case4 which are determined by the CLD.

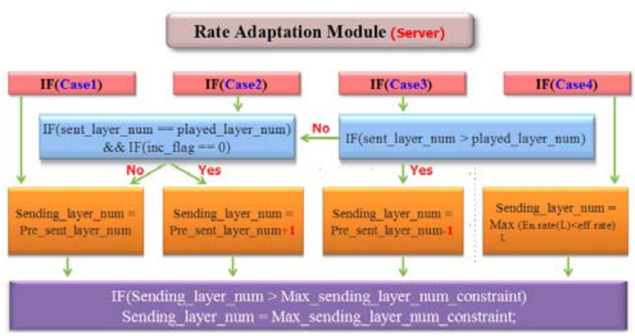


Fig.5 Rate adaptation algorithm

The Case1 represents the reasonable P_B and the excessive P_E. In this case, although reasonable P_B means that smooth BL transmission is being performed and more ELs can be transmitted, the excessive P_E informs the large amount of EL packets may lead the network congestion. Therefore, the rate adaptation module does not increase the sending layers and keep up the previous one. The Case2 represents both reasonable P_B and P_E. This means the rate adaptation module can increase the sending layers. However, abrupt change of sending rate may lead to network congestion. Therefore, the sending layers are slowly increased by using the Inc_flag that observes the transmission success. If the Inc_flag is equals to zero, it means the EL layers are transmitted without loss or late packet. The Case3 represents the reasonable P_B and the insufficient P_E. There can exist the two cases. The first case is that the server sends the several EL layers but the client does not receive every EL layers. In this case, the sending layers have to be decreased. The second case is that the server originally sends a little EL layers. In this case, the rate adaptation module increases the EL layers like the Case2. The Case4 stands for the heavy network congestion. Since the BL has a low encoding rate, the BL flow generally occupies a small portion of network resource. Therefore, the insufficient P_B means the current network has very low available bandwidth. In this case, although sending only BL packets can be useful solution, it does not best solution because the VBR-encoded video has constantly varying encoding rate. The proposed algorithm uses the effective receiving rate in the Case4 because it includes the information about the current network condition. The rate adaptation module selects the maximum layer number which does not excess the effective receiving rate. After the number of sending layers is determined about each case, the rate adaptation module readjusts the sending layers taking into account the upper constraint of sending rate which calculated by using GoP-based average encoding rate.

4. SIMULATION RESULTS AND PERFORMANCE EVALUATION

4.1 Simulation Conditions and Environments

The proposed algorithm is simulated with well-known ns-2 network simulator. The test video sequence is encoded by using the latest reference software JSVM 9.12.1 with the GoP size of 15, the frame type I/P and VBR. It is composed of a base layer and 4 enhancement layers. The BL is encoded at 107Kbps and the ELs are encoded from 264Kbps to 1897Kbps with the peak encoding rate of 7.2Mbps. The simulation lasts 240s and the standard video sequences of foreman (300 frames), hallmonitor (300 frames), coastguard (300 frames) and container (300 frames) are circularly used as the test video source.

Fig.6 depicts the simulation network topology. It has a bottleneck link between two routers (n1, n2) and all side link bandwidths are provisioned such that congestion will only occur at the bottleneck link. The bandwidth of the bottleneck link is set to 2.5Mb (10ms delay) and the side links bandwidth are set to 10Mb (5ms delay). The bottleneck link queueing policy is set to random early detection (RED).The start up delay is set to 2s and the feedback period is set to 200ms. To reduce server complexity, the server always sends both BL and EL layers of the same frame together.

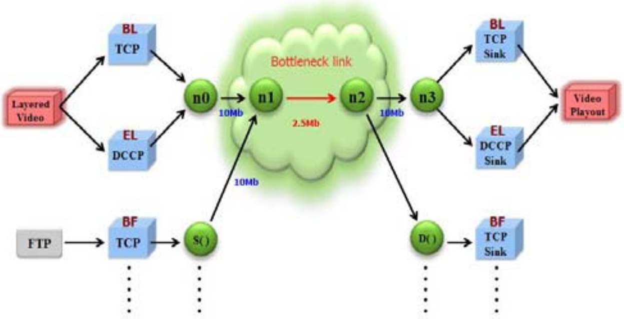


Fig.6 Simulation network topology

There are four simulation scenarios. The first simulation was performed without background TCP flows. The second simulation was performed with dynamically changed background TCP flows. The third simulation was performed with burst increasing background TCP flows. The fourth simulation was performed with burst decreasing background TCP flows. Fig.7 depicts the competing flows of each scenario.

Fig.7 Simulation scenarios

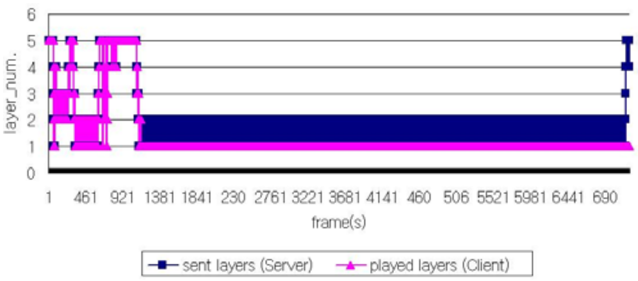
4.2 Simulation Results and Analysis

The end video quality is evaluated by PSNR, re-buffering count and ratio of sent/played layers. The ratio of sent/played layers is denoted as L_{Ratio}. The performance comparisons of the proposed algorithm and Gorkemli's algorithm are presented in Table 1 and Fig.8. In all the scenarios, we can show that the proposed algorithm provides the better performances. The L_{Ratio} of proposed algorithm is larger than Gorkemli's algorithm in all scenarios. It means that the proposed algorithm can better perform rate adaptation according to the congestion occurrence and alleviation. As a result, our algorithm could reduce the quality degradation that caused by lost/late packets between the server and the client.

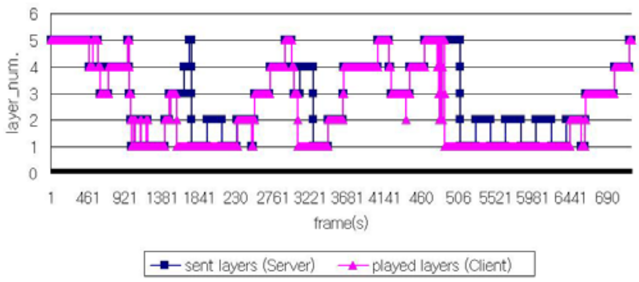
Table 1. Performance comparisons

		Avg.sent layers (A)	Avg.played layers (B)	L_{Ratio} (B/A)	Avg. PSNR	Rebuff. cnt.
Scenario #1	Gorkemli	3.04362	2.94680	0.9681	34.3300	1
	Proposed	3.69468	3.69468	1	35.9196	0
Scenario #2	Gorkemli	1.83776	1.38130	0.7516	30.6931	2
	Proposed	2.89054	2.52493	0.8735	33.2490	2
Scenario #3	Gorkemli	2.41825	2.13001	0.8808	32.4363	2
	Proposed	2.76496	2.57410	0.9309	33.3757	2
Scenario #4	Gorkemli	2.86289	2.38727	0.8338	32.9828	5
	Proposed	2.49145	2.09334	0.8402	32.3014	3

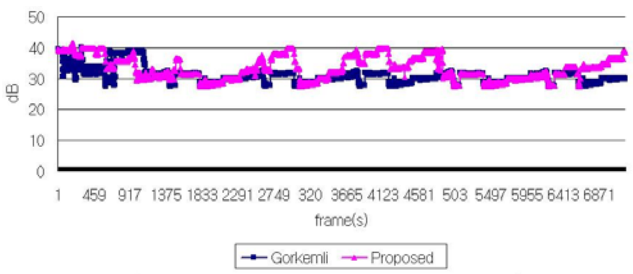
The simulation results of scenario #2 depict Fig.8. Fig.8(a) and Fig.8(c) show that the Gorkemli's algorithm may lead to large quality degradation by late packets in the case of dynamic background flows and VBR-encoded video sequence. On the other hand, Fig.8(b) shows that the proposed algorithm reduces the sending rate before the network congestion is occurred. In this way, the late packets could be prevented. As a result, our algorithm yields a PSNR of 33.24904 dB, which is 2.55589 dB higher than the 30.69315 dB result of Gorkemli's algorithm.



(a) Sent / played layers (Gorkemli's algorithm)



(b) Sent / played layers (Proposed algorithm)



(c) avg. PSNR

Fig.8 Performance comparisons (scenario #2)

5. CONCLUSION

In this paper, we proposed PoB-based rate adaptation algorithm to improve streaming quality. Since rate adaptation failure may lead to large quality degradation by late packet, it is important to detect the network congestion and determine the sending rate suited to network condition. The proposed algorithm tried to perform proper rate adaptation according to network condition by using the PoB estimator, CLD and GoP-based average encoding rate calculator. As a result, the simulation results showed that the proposed algorithm could provide better performance than pervious Gorkemli algorithm in terms of the average PSNR, L_{Ratio} and re-buffering count.

6. REFERENCES

[1] H. Schwarz, D. Marpe and T. Wiegand, “Overview of the Scalable Video Coding Extension of the H.264/AVC Standard”, IEEE Transactions on Circuit and System for Video Tech., Vol. 17, No. 9, September

[2] A. E. Mohr, E. A, Riskin and R. E. ladner, “Unequal Loss Protection: Graceful Degradation of Image Quality over Packet Erasure Channels Through Forward Error Correction”, IEEE Journal on Selected Areas in Communications, Vol. 18, No. 6, June 2000

[3] Y. Liu and O. Yu, “Adaptive unequal loss protection for scalable video streaming over IP networks”, IEEE Trans. Consumer Electronics, Vol. 51, No. 4, November 2005.

[4] S.-R. Kang and D. Loguinov, “Impact of FEC overhead on scalable video streaming,” NOSSDAV’05, pp. 123-128 June 2005

[5] RFC896, ‘Congestion control in IP/TCP internetwork’

[6] B. Gorkemli and M. Reha Civanlar, “SVC Coded Video Streaming over DCCP”, 8th IEEE International Symposium on Multimedia (ISM'06), 2006

Acknowledgement

This work was supported by KMoCIE (Korean Ministry of Commerce, Industry and Energy) and KETI (Korean Electronics Technology Institute) under project number 2008-8-0906.