ICAT 2003
December 3-5, Tokyo, JAPAN

ZEUS : Network Aware QoS Management for Immersive Environment*

Jin Ryong Kim[1], Yuichi Iwadate[2], Youjip Won[1]

[1] Div. of EECE, Hanyang University
17 Hangdang-dong, Sungdong-ku, Seoul, Korea
{ jessekim | yjwon }@ece.hanyang.ac.kr

[2] NHK Science & Technical Research Laboratories, Japan Broadcasting Corporation
1-10-11 Kinuta, Setagaya-ku, Tokyo, Japan
iwadate.y-ja@nhk.or.jp

Abstract

The production value accomplished with virtual studio applications is becoming more attractive to the modern broadcasting studios. One of the drawbacks in today's virtual studio is the limitation of viewpoint. Virtual studios using multiple cameras can overcome the restriction of viewpoint. To efficiently deliver the continuous media(CM) from a studio to viewers, we must consider the heterogeneous, irregular network environments. In this paper, we focus on the guarantee of constant playback in the IP network. We present the network aware QoS management for immersive environment to deliver the media in time and at the same time, to minimize the quality variation. We propose the adaptive QoS management scheme to minimize the quality variability while at the same time maximize the utilization of variable network bandwidth. As a result, the stable frame transmission rate is guaranteed and the quality fluctuation becomes smoother.

Key words: QoS, Virtual Studio, Smoothing, Interactive TV, Multimedia Streaming, VRML

1. Introduction

1.1 Motivation

Recently intensive research activities to overcome the limit of viewpoint in the TV contents are increasingly in progress. One of the solutions to overcome the restriction of viewpoint is to use multiple cameras for generating 3D model images in the virtual studio. By using multiple cameras in a virtual studio, TV audiences can enjoy the media contents in any viewpoints. Fig. 1 shows the general idea of our system. In the broadcasting center, a dancer surrounded by multiple cameras is dancing in a virtual studio. The images of the dancer are continuously captured from the cameras and 3D model is synthesized for each frame. The broadcasting center then starts to transmit 3D model frames to the TV audiences through the network. The audiences use TV, PC, or PDA to watch the dancer's performance in any viewpoint.

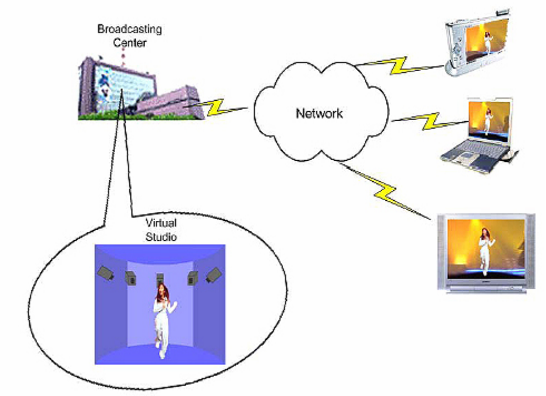

Fig. 1 General concept

Delivering media contents from virtual studio to the consumer platforms represents a particularly interesting domain for this subject. When the high-resolution, three-dimensional, time-dependent media contents move downstream to viewers, transmission rate in the network will be varied depending on the network traffic condition, which may cause unstable playback rate. Thus, we should consider the heterogeneous network variability to support streaming service efficiently to the audiences. As a matter of fact, distributed collaborative environments and video conferencing have dynamically changing and unpredictable QoS requirements. This problem is caused by the heterogeneous nature and varying capabilities of today's end-systems and network infrastructures. To solve this problem, the system infrastructure must be adaptive.

We propose an adaptive QoS management to efficiently stream time-critical media from a server to multiple heterogeneous clients across a heterogeneous network.

* This work is funded by KOSEF grants number R11-2000-073-01000-0.

In this paper we make two main approaches for QoS management: adaptive QoS management scheme and optimal quality adaptation.

1.2 Related Work

NHK[1] developed a 3D model generation system using multiple cameras with multi-baseline stereo algorithm and the volume intersection method to overcome the restriction of viewpoint. In this system, VRML animation frames are generated from images of 300 continuously captured frames. Nine cameras are used to surround the target object and operated simultaneously. The object is a moving human body and the 3D model is generated from images of the object captured from multiple cameras. As other studies, CMU developed a modeling system by using 51 cameras to obtain a moving human model[2]. Their system uses a multi-baseline stereo algorithm to acquire a human body model and image-based rendering to generate video from any viewpoint. Kyoto University also developed a modeling system from multiple camera images[3], using a volume intersection method by which an approximate shape of the object is obtained.

The efficient scalable video coding has been established under the Amendment on Streaming Video Profile of the MPEG-4 standard[4]. Salehi[5] proposed an optimal rate smoothing algorithm based on the traffic smoothing technique to achieve minimum variability of transmission rate. Optimal quality adaptation algorithm based on scalable video has been also proposed in [6]. The authors in [6] proposed to find a shortest path to minimize variability. The authors in [7] accomplished the maximum reduction of quality variability for layered CBR video using bidirectional layer selection. The authors in [8] proposed quality adaptation algorithm for VBR video using state transition mechanism.

This paper extends NHK's multiple cameras system by developing the QoS architecture to support the network environments. We developed streaming system for immersive environment using frames captured from NHK's multiple cameras system. It is designed for delivering visual contents from the studio to the consumer platform.

2. Adaptive QoS Management

2.1 Problem Formulation

Delivering 3D media via a best effort network exhibits some of the challenging aspects since the available bandwidth fluctuates over a broad range. This makes difficult to provide perceptually good quality of streaming video. A key challenge in delivering multimedia data across the Internet is to reduce delays and make efficient use of the network bandwidth. In the IP network, there is no priority guarantee of the connection to reserve the bandwidth in the link and the network traffic variance is hard to predict. Under these circumstances, it is difficult to guarantee to transport time-critical media contents in time. One of the solutions is to provide scalable encoding scheme depending on the network traffic condition. In this approach, we carefully argue that the larger size of the contents takes more time to transport than the smaller size.

We propose scalable 3D media encoding scheme. Continuous media stream in this approach consists of frames. By differently reducing the numbers of polygons in each frame, we generate several different encoded 3D videos. In this approach, we divide them into five quality levels. There is no restriction on how many categories we need to have. It is based on the network condition: subscribing to as many scalable videos as possible when the available bandwidth is large, and dropping scalable videos when the available bandwidth is small. The reason for dividing into five categories is because we believe that it is appropriate for the present network bandwidth variability and quality variation.

We define QoS levels. QoS levels are divided into five levels and each level 'E', 'G', 'F', 'P', and 'B' are denoted in terms of 'Excellent', 'Good', 'Fair', 'Poor' and 'Bad'. It is necessary to make sure that the better quality of frame has larger size. Therefore, 'Excellent' quality of video has the largest data sizes and 'Bad' quality video has the smallest data sizes. This scheme can be highly adaptable to the Internet's bandwidth fluctuation by distributing the data over a wide range of bit rates. We consider this quality adaptation algorithms based on stored video for a unicast environment.

2.2 Rate Adaptive Transmission

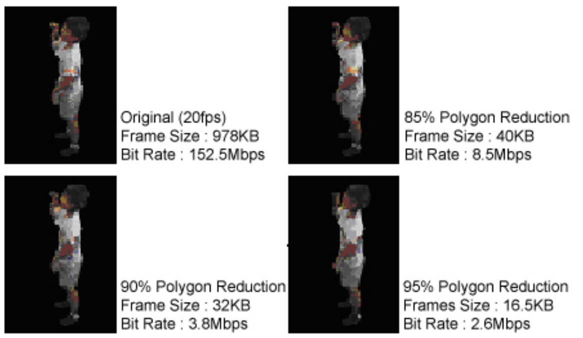

Fig. 2. Reduction of number of polygons

Original video consists of 300 frame sequence and is a bit rate of 152.5Mbps. It is CBR encoded at 20 fps and a typical frame size is 976KB. Since the data size is very heavy, it is not appropriate to distribute them into the high speed network. In Fig. 2, we separate the original 3D movie into five different movies by differently changing the reduction ratio of their number of polygons. It is reduced by 85%, 87%, 90%, 93%, and 95% and mapped into 'E', 'G', 'F', 'P', and 'B', respectively. We reduce the number of polygons from the original movie and reproduce five, differently encoded movies. Each movie has different bit rate under the same frame rate. The idea of our scheme is to monitor the network bandwidth availability and send the movie having appropriate bit rate. Each time slot represents the time unit for playing a movie. For simplicity, we assume that the current network bandwidth availability is known a priori. We also assume that the startup delay of one time slot and we ignore the network delay. Table 1 summarizes the notation.

Table 1 Notation

t : point of time
k : time slot at t_k
$W(k)$: Available network bandwidth
ϕ_k : Inner portion of network bandwidth curve
δ_k : Outer portion of network bandwidth curve
C : Number of frames in a time slot
$QoS(k)$: QoS level at k
$Q(k)$: Quality value mapped to $QoS(k)$
$X(QoS(k))$: Bit rate for $QoS(k)$

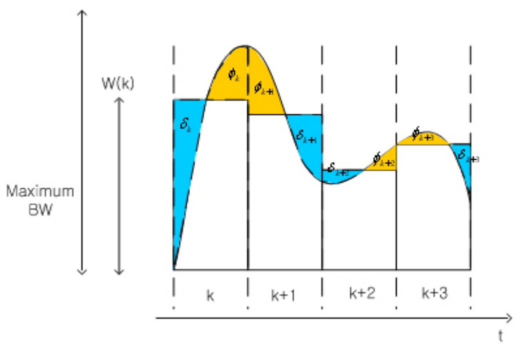

Fig 3 Available network bandwidth

Let t be the point of time and k be the time slot at t_k. We set C=20 for every time slot so that we have fixed frame rate for every time slot, which is 20 fps. We assume that the current *W(k)* is known and it is illustrated in Fig. 3. In Fig. 3, we define *W(k)* as a quantitative amount of the available network bandwidth at each time slot *k*. *W(k)* is an optimal value at each time slot *k* from the network available bandwidth curve. This optimal value is set under $\delta_k \geq \Phi_k$ for all time slots. Thus,

$$QoS(k)=\begin{cases} \text{E, if } W(k) \geq X(E) \\ \text{G, if } X(E) > W(k) \geq X(G) \\ \text{F, if } X(G) > W(k) \geq X(F) \\ \text{P, if } X(F) > W(k) \geq X(P) \\ \text{B, otherwise} \end{cases} \quad (1)$$

2.3 Viewpoint Control

In this section, we focus on content-aware transport using viewpoint information. Viewpoint based QoS management scheme has been proposed in the TIE system[9]. The QoS level is determined by the distance between observer's and object's positions in the virtual space. By adjusting the frame rate of the object's video stream mapped to its QoS level, the overall network traffic can be reduced. In our work, we adjust the rate adaptation to reduce the unnecessary data using the distance between object and observer. The user observes the object at the viewer's position. The object near the observer's view point looks more clear and larger in the client screen. The QoS level of the object depends on the distance from the observer's view point. Let (x, y, z) and (x', y', z') be the object's and the viewer's position, respectively in three dimensional space. The distance D between the object's position and the viewer's position is computed as in Eq. 2.

$$D = |z' - z| \quad (2)$$

We need to obtain the QoS level for the object based on the distance D in Eq. 2. For this purpose, we introduce the QoS mapping function as in Eq. 3.

$$f(D) = \frac{1}{D^{1/m}} \quad (3)$$

The object's QoS level is inversely proportional to the square of the distance between the object and the viewer. m is the constant value that adjusts the slope of the mapping curve. f maps the distance to one of the set of integers in $\{1,\ldots,n_{Zoom}\}$. n_{Zoom} is the number of distinct zoom levels provided in the system and it corresponds to 3 in our system. Thus, *Q(k)* can be written as

$$Q(k)=\begin{cases} Q(k) \text{ if } f(D) = Normal \\ Q(k)+1 \text{ if } f(D) = In \\ Q(k)-1 \text{ if } f(D) = Out \end{cases} \quad (4)$$

3. Optimizing Quality Variation

3.1 Smoothness Criterion

It is generally agreed that significant quality fluctuation caused by frequent changing the video quality may be annoying and degrade the perceptual quality of video. The problem of rate adaptive transmission is the frequent changes of video quality due to the variation of the network bandwidth. As the objective of the previous scheme is to maximize the utilization of network, it may incur significant quality variation.

In this section, we propose an optimal schedule scheme to minimize the quality variability while at the same time maximize the utilization of variable network bandwidth. Fig. 4(a) illustrates the unstable video transmission if we transmit only one quality scale of the video sequence and Fig. 4(b) illustrates how the video sequence is adaptively transmitted under variable network bandwidth using our rate adaptive transmission scheme. Fig. 4(c) illustrates the quality fluctuation. Each colored time slot represents the distinct quality. The basic idea of our extended scheme is that instead of changing the quality of the time slot at each point of time, we keep the same quality and raise the quality level at some point later. We accomplish this scheme by prefetching some portions of the next time slot as shown in Fig. 4(d). This enables us to maximize the usage of available network bandwidth and minimize the quality variation.

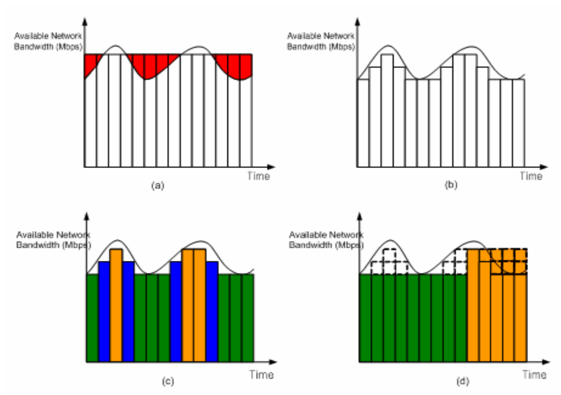

Fig. 4 Example of quality optimization

The question is how to determine the quality transition point. The author in [8] used MA(Moving Average) type estimator to determine the available bandwidth. Since we know the current available network bandwidth, we forecast the future network bandwidth availability using double exponential smoothing based predictor (DESP)[10]. DESP is one of very popular schemes to produce a smoothed Time Series. Whereas in MA the past observations are weighted equally, DESP assigns exponentially decreasing weights as the observation get older and considers a trend. DESP is approximately 135 times faster than Kalman and extended Kalman filter-based predictors with roughly the same accuracy[11]. Using DESP, the future available network bandwidth can be forecasted as in Eq. 5.

$$P(k) = \alpha W(k) + (1-\alpha)(W(k-1) + b(k-1)) \qquad (5)$$
$$b(k) = \gamma(P(k) - P(k-1)) + (1-\gamma)b(k-1)$$

where *P(k)* is smoothed value at k and *b(k)* is trend equation. α and γ are smoothing and trend constants, respectively, and $\alpha \in [0,1]$ and $\gamma \in [0,1]$. Forecast equation *Z(k)* is defined as

$$Z(k+1) = P(k) + b(k) \qquad (6)$$
$$Z(k+u) = P(k) + ub(k)$$

where u denotes *u-period* ahead forecast.

3.2 Framework

We propose a formal model for quality smoothing and describe how to provide smoother quality layered stream. We present layer based smoothing algorithm with stochastic method. Layer based smoothness criteria is proposed by Nelakuditi's work[7]. We model the quality adaptation for CBR video by replacing re-scheduled time slots to maintain a uniform quality.

Fig. 5 illustrates the framework of optimal quality adaptation. In formulating optimal quality adaptation, we consider a discrete-time model. We assume that there are 5 seconds of buffers. The time slots in the primary scheduler are based on DESP and the secondary scheduler is for rescheduled time slots. The primary and secondary buffers in the client are for the time slots in the primary and secondary schedulers, respectively. Previously we assumed that the startup delay of one time slot so that the server starts video transmission one time slot ahead of the time the client starts buffering. We set u as 4 in Eq. 6 to have four time slots in the primary scheduler.

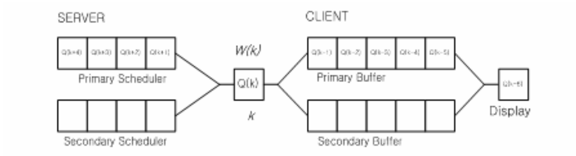

Fig. 5 Framework of server and client

We consider the future available network bandwidth trend as a reference quality scale for every point of time. Let i be the index and N be the number of time slots in the primary scheduler, which can be replaced by u. Then, the mean quality scale of the time slots in the primary scheduler is

$$A = \left\lfloor \frac{\sum_{i=k+1}^{N+k} Q(i)}{N} \right\rfloor \qquad (7)$$

Note that we take a floor in Eq. 7 because the quality is scaled in integer and it cannot be rounded off.

We now address the question of how to select an optimal quality scale. To accomplish this, we consider a dominant quality scale in the system. That is, we find the dominant quality scale and reschedule other time slots with it to maintain an even quality scale. Fig. 6 illustrates the selection of a dominant quality scale. We define three states in the system: *future*, *present*, and *past*. In the *future state*, the time slots are scheduled based on DESP and they can be rescheduled with other quality scales. In the *present state*, the time slots are still unplayed and they are not met the deadline for display. In another word, those in the *present state* have chances to be replaced with other quality scales. However, the time slots in the *past state* are not targeted for rescheduling due to the limit of time. That is, those in the primary buffer in the *past state* should be played out within 2 time slot intervals but the rescheduled time slots need 3 time slot intervals for display. Thus we only consider the *future* and *present states* for rescheduling time slots with a dominant quality scale. Let us look at Fig. 6(a). In this figure, several quality scales are arranged in the system. We notice that the quality level of 'F' is a dominant quality scale in the system as in Fig. 6(b). Let $f_R(r,t)$ denote a set of frequencies of quality scales where $r \in \{E, G, F, P, B\}$. Then, the dominant quality scale is

$$\text{Dominant} = \max\{f_R(r,t)\} \tag{8}$$

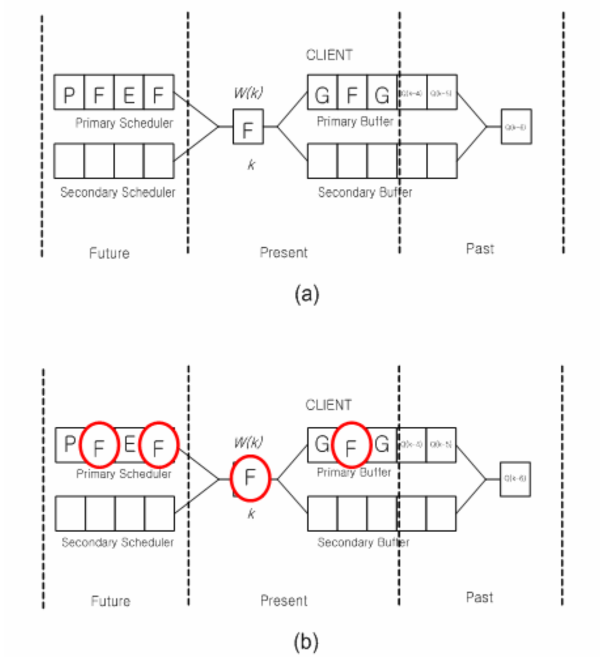

Fig. 6 Dominant quality scale

3.3 Prefetching Algorithm

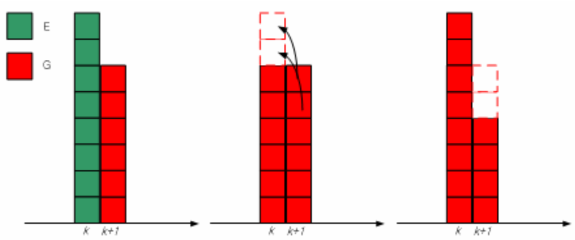

Fig. 7 Prefetching granules

We develop a prefetching algorithm. The prefetching algorithm plays a key role in our architecture. In the following, we show how to prefetch the portions of untransmitted time slots and build its algorithm. The basic idea of prefetching is illustrated in Fig. 7. In this figure, the time slots are divided into layers with the same size and it is represented as granules. When we transmit the time slot with rescheduled quality scale, some vacant granules become available. Then, we take advantage of using these empty spots to prefetch some granules from the next time slot. For example, let the quality scale of k and k+1 be 'E' and 'G', respectively. If we decide to lower the quality level of the time slot k to 'G', then two empty granules become available. Then, we prefetch two granules from the time slot k+1 into the empty spots in the time slot k. We keep prefetching granules until we save enough to upgrade the quality.

In order to accomplish this, we must keep track of residual bandwidth *RB(k)* and it can be calculated as

$$RB(k) = RB(k-1) + W(k) - X(\hat{k}) \tag{9}$$

where $\hat{k}$ is a rescheduled time slot. Fig. 8 shows the procedure for prefetching algorithm. In Fig. 8, *L(k)* denotes the depth of layer at time slot k and *e(k)* denotes the number of empty granules. l and j denote the size of granule and the layer index, respectively. Using *L(k)* and *e(k)*(line 2-3), we prefetch granules in $k+1$(line 4).

1. PROCEDURE PREFETCH($k+1$, $RB(k)$)
2. $L(k) = \dfrac{X(k)}{l}$
3. $e(k) = \dfrac{RB(k)}{l}$
4. Prefetch= $\displaystyle\sum_{j=L(k+1)-e(k)+1}^{L(k+1)} Q^j(k+1)$
5. END PROCEDURE

Fig. 8 Prefeching algorithm

3.4 Optimal Quality Adaptation Algorithm

We present an optimal quality adaptation algorithm using DESP, mean and dominant quality scale, and prefetchng algorithm. The purpose of this scheme is to minimize the quality variation by rescheduling time slots and maximize the utilization of network bandwidth as we proposed in the previous section.

Fig. 9 shows the details of this algorithm. It consists of two stages: one in initial stage(line 2-3) and one in scheduling stage(line 4-15). During initial stage, the point of time and residual bandwidth are initialized.

In scheduling stage, we schedule the time slots in a primary scheduler using DESP in Eq. 5 and 6 and find a reference quality scale(line 5). Then we find a dominant quality scale(line 6). By comparing two values, we select optimal scale(line 7-12). If the dominant scale in the system is the same as a reference quality scale, we select either dominant or reference quality scale without hesitation(line 7-8). However, if the mismatch occurs, we choose a dominant quality scale if the residual bandwidth is enough for transmission(line 9-10). However, if the residual bandwidth is not enough for delivering rescheduled time slot, then we proceed with the reference quality scale since $A \leq$ Dominant(line 11-12). Next, residual bandwidth is calculated and we prefetch the next time slot(line 13-14).

1. PROCEDURE$(k,\ W(k))$
2. Initialization
3. $t = 0,\ RB = 0$
4. Scheduling
5. Scheduler: $A = \left\lfloor \dfrac{\sum_{i=k+1}^{N+k} Q^j(i)}{N} \right\rfloor$
6. Dominant $= \max\{f_R(r,t)\}$
7. IF(Dominant==A)
8. $Q(k+1) = A$
9. ELSE IF $RB(k) \geq X(\widehat{k})$
10. $Q(k+1) =$ Dominant
11. ELSE
12. $Q(k+1) = A$
13. $RB(k) = RB(k-1) + W(k) - X(\widehat{k})$
14. PREFETCH$(k+1,\ RB(k))$
15. Update t
16. END PROCEDURE

Fig. 9 Optimal quality adaptation algorithm

4. EVALUATION

4.1 Experimental Design

We conduct experiments to study the performance of rate adaptive scheme and optimal quality adaptation algorithm. We first demonstrate the performance of our proposed algorithm by comparing with the system where our scheme is not applied. Then, we evaluate the performance of our optimal quality adaptation algorithm and compare it to the rate adaptive scheme using QTD(Quality Transition Distribution) and ARL(Average Run Length) [7]:

$$QTD = \frac{b}{Y} \qquad (10)$$

$$ARL = \frac{1}{L}\sum_{i=1}^{L} \frac{\sum_{j=1}^{k_i} n_j}{k_i}$$

where b is the number of quality transition and Y is the number of time slots in the entire movie. k_i is the number of runs in the ith layer, and n_j is the length of the jth run. The experiment is set under local network. We implement the server and client system and we place the network traffic generator and receiver to share the link. We intentionally increase the traffic rate in the link to see how it influences the system and how our proposed schemes adaptively manage the streaming service.

4.2 Performance of Rate Adaptive Transmission

In this experiment, we demonstrate the performance of our rate adaptive transmission scheme by comparing with system that our scheme is not applied. We observe the results consequent to our experimental design and validate the effectiveness of our proposed QoS management scheme. The network traffic generator is UDP-based and it disturbs the streaming service by sending dummy packets to the server.

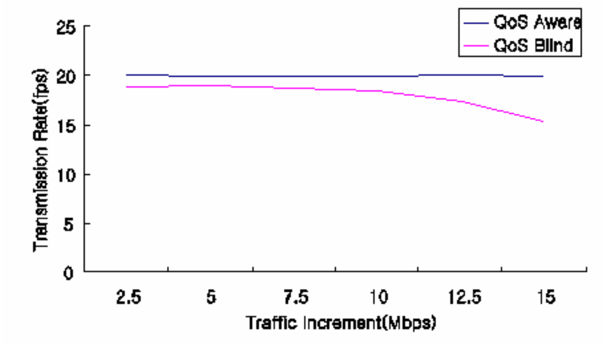

Fig. 10 Performance of rate adaptive transmission

Fig. 10 illustrates the effectiveness of our proposed rate adaptive transmission scheme. In this figure, we increased the network traffic up to 15 Mbps. When we apply our rate adaptive transmission scheme on our system, frames are transmitted within constant rate, which is 20 fps, under irregular network traffic condition. On the other hand, the frame rate is decreased and not stable in the system where our scheme is not applied. When the network traffic reaches 15 Mbps, the transmission rate difference between two systems becomes 5 fps, which results 0.2 seconds delay per frame. This result shows that it causes the rate fluctuation under varied network bandwidth circumstances. Without using scalable encoded videos, it is not tolerated to viewers under irregular network environments. This result proves that our rate adaptive transmission scheme can guarantee the constant frame rate in an irregular network environment.

4.3 Performance of Optimal Quality Variation

We now investigate the effectiveness of our smoothing algorithm. We measure the performance in terms of the QDT and the ARL. Fig. 11 illustrates the performance of rate adaptation transmission scheme in each traffic increment interval. Fig. 11(a) through (d) illustrate the quality fluctuation with the traffic interferences at 2.5, 5, 10, and 15 Mbps, respectively. In Fig. 11(a), QTD is 0.78 and ARL is 1.282. The quality transitions are occurred mostly among ‘E’, ‘G’, and ‘F’. In Fig. 11(b), QTD and ARL are 0.8 and 1.263, respectively. Note that the dominant quality scale in this figure is ‘F’ and some transitions are occurred between ‘G’ and ‘F’. In Fig. 11(c) and (d), QTDs are 0.63 and 0.67 at 10 and 15 Mbps. ARLs are 1.587 and 1.493. As results, the rate adaptation transmission scheme shows the quality fluctuation and degradation of perceptual quality due to its poor quality adaptation.

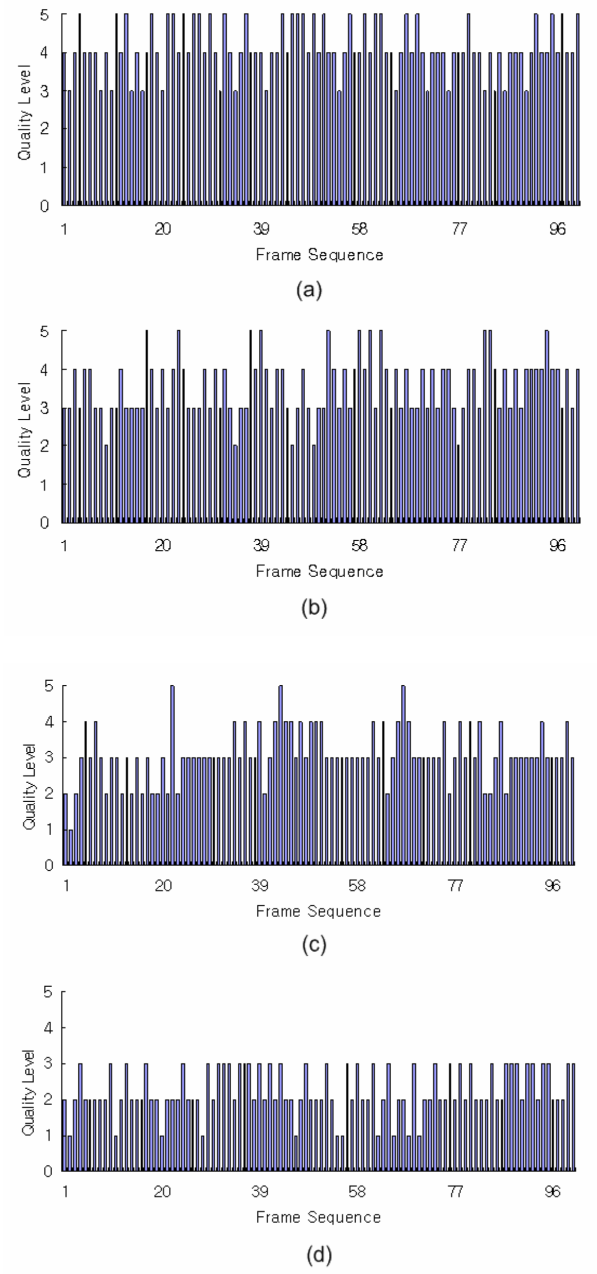

Fig. 11 Performance of unsmoothed streaming

Fig. 12 illustrates the performance of optimal quality adaptation algorithm under the same conditions as the rate adaptation scheme. We set the experiment under the same circumstances as Fig. 11 using optimal quality adaptation. We set $\alpha=0.5$ and $\gamma=0.28$. We compare the quality variability using optimal quality adaptation algorithm with unsmoothed, rate adaptation transmission scheme in Fig. 11. In Fig. 12(a), QTD is 0.13 and ARL is 7.846. Note that QTD is smaller and ARL is longer in optimal quality adaptation than those in rate adaptation scheme. In Fig. 12(b), QTD and ARL are 0.07 and 14.286, respectively. In this figure, it is noticed that only two quality scales are used and an extremely smoothed result is accomplished. In Fig. 12(c), QTD is 0.08 and ARL is 12.25. ARL is definitely longer in optimal quality adaptation and it brings smoother quality variation, compared to Fig. 11(c). In Fig. 12(d), QTD and ARL are 0.1 and 8.909. Overall, the results show that the optimal quality adaptation algorithm exhibits smoother quality fluctuation compared to the rate adaptive transmission scheme.

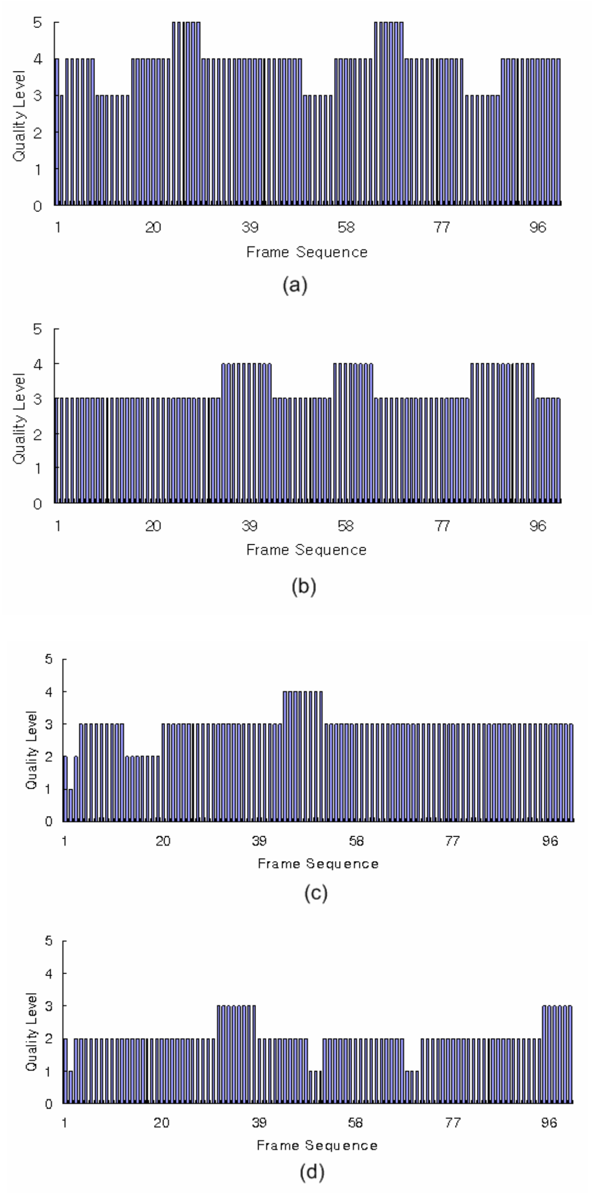

Fig. 12 Performance of smoothed streaming

We measure the correctness of the network available prediction. We use DESP over typical network throughput and see how close they are fitted using the sum of squared error(SSE) and the mean squared error(MSE). Fig. 13 illustrates the performance of DESP. Fig. 13(a) shows the performance when the prediction takes only one-period-ahead(u=1) and Fig.

13(b) shows the performance when the 3-periods-ahead is forecasted(u=3). In Fig. 13(a), the SSE and the MSE are 361.44 and 3.698, respectively. In Fig. 13(b), the SSE is 648.73 and the MSE is 6.829. The result in Fig. 13(b) is almost double as Fig. 13(a), but still they are closely fitted.

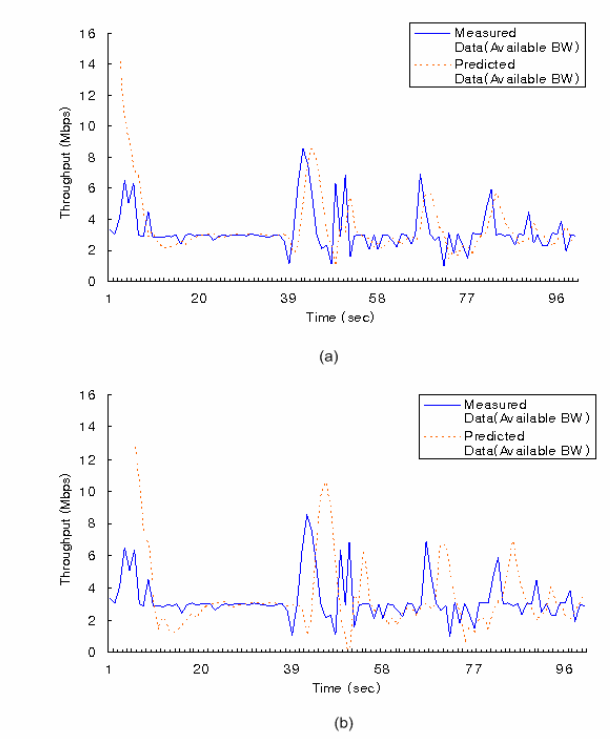

Fig. 13 Performance of DESP

5. CONCLUSITION

In this paper, we proposed adaptive QoS management to reduce time delay and guarantee the constant playback rate in delivering high-quality, 3D CM contents. We also provided optimal quality adaptation scheme to minimize the quality fluctuation. Our results show that proposed QoS architecture can effectively utilize the available network bandwidth and minimize the quality variation.

A couple of issues should be elaborated on further and have been under our investigation. First, we focus on the relevance of frames in the client to generate higher qualities by using frames in the past history. When the network bandwidth is changed from high to low, we reduce the number of polygons of 3D images to make it smaller. However, if we use the past frames, which transmitted under good qualities, we have a good chance to use their relevance for re-modeling the low quality frames in the future. Another issue that we concern about is the heterogeneous, multiple clients in the heterogeneous network environment. To support the stable streaming service to the viewers, variety of consumer platforms, wire or wireless, in the heterogeneous, irregular network environment must be under consideration.

ACKNOWLEDGEMENT

We would like to thank Dr. Miwa Katayama, Mr. Kimihiro Tomiyama and Mr. Masaki Takahashi in NHK-STRL for their work on the 3D model system.

References

[1] Y. Iwadate, M. Katayama, K. Tomiyama, and H. Imaizumi, "VRML Animation from Multi-view Images," *Proceedings of ICME2002 IEEE International Conference on Multimedia and Expo*, pages 881-884. IEEE, August 2002.

[2] P. Rander, "A Multi-camera Method for 3D Digitization of Dynamic Real-world Events," *Doctoral Dissertation: CMU-RI-TR-98-12*, pages 1-119. Robotics Institute, Carnegie Mellon University, May 1998.

[3] T. Wada, X. Wu, S. Tokai, and T. Matsuyama, "Homography based Parallel Volume Intersection: Toward Real-time Volume Reconstruction using Active Cameras," *5th International Workshop on Computer Architecture for Machine Perception*, pages 331-339. IEEE, September 2000.

[4] *Coding of Audio-Visual Objects, Part-2 Visual, Amendment 4: Streaming Video Profile*, ISO/IEC 14496-2/FPDAM4, July 2000.

[5] J. D. Salehi, Z.-L. Zhang. J. Kurose, and D. Towsley, "Supporting Stored Video: Reducing Rate Variability and End-to-end Resource Requirements through Optimal Smoothing," *IEEE/ACM Trans. Networking*, vol. 6, no. 4, Aug. 1998.

[6] P. de Cuetos and K. W. Ross, "Adaptive Rate Control for Streaming Stored Fine Grained Scalable Video," *Proceedings of NOSSDAV 2002*, Miami, FL, May 2002.

[7] S. Nelakuditi, R. R. Harinath, E. Kusmierek, and Z.-L. Zhang, "Providing Smoother Quality Layered Video Stream," *Proceedings of NOSSDAV 2000*, Chapel Hill, NC, June 2000.

[8] T. Kim and M. H. Ammar, "Optimal Quality Adaptation for MPEG-4 Fine-Grained Scalable Video," *22nd Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM 2003)*, IEEE, Volume: 1 , March - April 2003 Page(s): 641 -651.

[9] J. R. Kim, Y. Won, B. Kim, "TIE: Networked Virtual Environment for Heterogeneous Environment", *Proceedings of International Conference on CIC*, October 28~31, 2003, Seoul, Korea, pages 534~538.

[10] C. Chatfield, *Time-Series Forecasting*. Chapman and Hall/CRC, 2001.

[11] J. J. LaViola Jr., "Double Exponential Smoothing: An Alternative to Kalman Filter-Based Predictive Tracking," International Workshop on IPT/EGVE 2003, Zurichi, Switzerland, May 2003.