MPEG-4 VIDEO STREAMING WITH DRIFT-COMPENSATED BIT-STREAM SWITCHING

Yeh-Kai Chou, Li-Chau Jian, and Chia-Wen Lin

Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi, Taiwan 621, ROC
Email: cwlin@cs.ccu.edu.tw

ABSTRACT

Bandwidth variations is one of the major problems in providing QoS guaranteed services for the current Internet users. The FGS coding scheme is considered an efficient and effective solution to solving the bandwidth variation problem. However, its coding efficiency is significantly lower than the non-scalable coder when the supporting bit-rate range is wide. Streaming video with multiple bitstream switching may provide a more efficient solution for rate adaptation for users with a large degree of heterogeneity than using a single bitstream. In this paper, we propose a receiver-driven drift-compensated switching method for rate adaptation in heterogeneous video multicasting applications. Our proposed system can achieve significant performance improvement in case of packet loss or network congestion.

1. INTRODUCTION

With the proliferation of online multimedia content (e.g., electronic documents, emails, news, entertainment and commercial video clips, etc.), the popularity of multimedia streaming technology, and the establishment of video coding standards, people are able to ubiquitously access and retrieve various multimedia contents via the Internet, promoting networked multimedia services at an extremely fast pace. Real-time multimedia, as the name implies, has timing constraints. For example, video sequences must be played out frame by frame. If the client cannot receive the video data in time, the playout process will pause or collapse, and then annoys the human visual system. In addition to the time constraint, video streaming over the Internet presents several challenges, such as high storage-capacity and throughput in the video server and high bandwidth in the network to deliver a large number of video streams.

Bandwidth variation is one of the primary characteristics of “best-effort” networks, and the Internet is a prime example of such networks. Scalable coding plays a crucial role in delivering the best possible quality over unpredictable “best-effort” networks by trading the streaming video quality with the changing network conditions. The newly established MPEG-4 Fine-Granularity Scalable (FGS) [5], has several outstanding features of low complexity, supporting a wide range of user bit-rates rather than some discrete rates, and packet-loss resilience, making it especially suitable for video streaming applications. It thus has been adopted as the coding tool in the MPEG-4 streaming video profile [5]. However, the benefits come with the cost of poorer coding efficiency, especially when the bit-rate rage is wide. As a result, a single FGS bitstream may not achieve best tradeoff among cost, flexibility, and performance for users with different requirements and capacities. Therefore streaming video with multiple FGS bitstreams has been proposed to serve users with a large degree of heterogeneity [1-3]. However, direct switching from one bitstream to another directly will result in drifting errors, which can lead to significant quality degradation.

In this paper, we present a receiver-driven congestion control scheme which can support video multicasting with multiple FGS bitstream switching. Our proposed system can estimate the TCP throughput by detecting the packet loss ratio and measuring the round-trip delay values in a receiver-driven manner, and then perform the congestion control for heterogeneous video multicasting accordingly. Besides, in order to serve users who may have fluctuated bandwidth such as modem or ADSL, we proposed a drift-compensated bitstream switching scheme to reduce the drift error when performing switching for rate adaptation. Furthermore, by combining the receiver-driven multicasting mechanism and the proposed multiple bit-stream switching, our proposed system can achieve significant performance improvement in case of packet loss or network congestion.

The rest of this paper is organized as follows. In Section 2, we present the drift-compensated bitstream switching algorithm. Section 3 describes the proposed MPEG-4 video streaming system with multicasting capability. In addition, a receiver-driven multiple bitstreams switching is for heterogeneous video multicasting is also proposed. The experimental results and conclusions are shown in Sections 4 and 5, respectively.

2. DRIFT-COMPENSATED BITSTREAM SWITCHING

In streaming video applications, the server may provide several bitstreams with different bitrates for ech client to switch over the bistreams to choose the bitstream which matches the client's channel bandwidth the most for rate adaptation. For instance, clients with high channel bandwidths can subscribe to higher-rate bitstreams for better video quality, while low-bandwidth clients need to subscribe to lower-rate bitstreams with worse perceptual visual qualities. There are some issues with such bitstream switching we have to concern about. When the available channel bandwidth drops, clients have to switch from one higher-rate bitstream to another lower-rate one (a "switching-down" process), and vice versa. Both the switching-up (from a lower-rate to a higher-rate) and switching-down (from a high-rate to a lower-rate) processes will introduce drifting errors as will be explained below.

2.1. Multiple Bitstream Switching with Drift Compensation

Our proposed algorithm mainly focuses on the encoder side. For simplicity but without loss of generality, we assume that the streaming video server just provides two bit-streams of different bit-rates (can be extended to more than two bitstreams) for clients to choose. One is encoded at a higher bit-rate R_{h}, and the other is encoded at a lower bit-rate R_{l}. These bitstreams are offline-encoded and stored at the server side. Besides these two bitstreams, we propose to use two extra drift-compensation bitstreams based on the concept of "SP" frames in [3] and drift-compensation frames in [4].

The way how we encode the drift-compensation bitstreams goes as follows. Let H represent the higher-rate bistream, L represent the lower-rate one, D^{HL} be the drift-compensation bitstream used for switching from H to L, and D^{LH} be the drift-compensation bitstream used for switching from L to H. D^{HL} and D^{LH} are generated using the following equations.

$$D_n^{\mathrm{HL}} = \mathrm{Pred}(H_{n-1}, L_n) \tag{1}$$

and

$$D_n^{\mathrm{LH}} = \mathrm{Pred}(L_{n-1}, H_n) \tag{2}$$

where H_n and L_n stand for the nth frames of bitstreams H and L, respectively; Pred(A,B) represents an inter-frame prediction process that frame B is predicted from the reference frame A. Fig. 1 shows the process of generating the D^{HL} bitstream, where the drift-compensation frames D_n^{HL} are encoded using an MPEG-4 coder (without the IDCT and inverse quantizer) with L_n as the input and the H_{n-1} stored in the frame memory (instead of L_{n-1}) as the reference. The D^{LH} bitstream can be obtained using a similar manner, while H_n is used as the input and L_{n-1} is stored in the frame memory as the reference for prediction. Fig. 2 shows the temporal and prediction relationship among D^{HL}, H, and L.

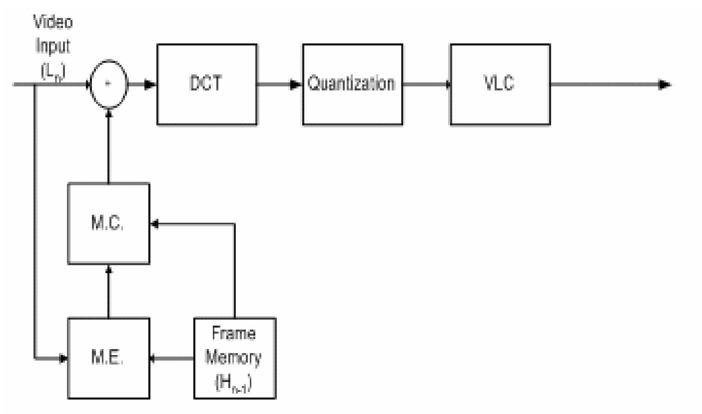


Fig. 1. An example of generating the D^{HL} bitstream.

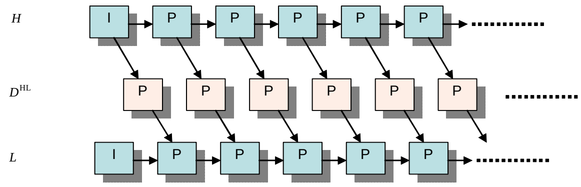


Fig. 2. Temporal and prediction relationship among D^{HL}, H, and L.

When a client requests the server to perform bitstream switching, the drift-compensation frames are used for switching between H and L. For example, if a switching down operation (i.e., switching from H to L) at frame n is requested, the server will send frames as …, H_{n-1}, D_n^{HL}, L_{n+1}, …, instead of …, H_{n-1}, L_n, L_{n+1}, … as illustrated in Fig. 3 with frame 4 as the switch-point. With our proposed algorithm, the server will send one drift-compensation frame to replace the frame at the switch-point in the new bitstream to switch to. In the example shown in Fig. 3, after sending frame 4 of H, the server sends frame 4 of D^{HL} (the switch-point) followed by frame 5 of L, leading to a sequence as $..., H_2, H_3$, D_4^{HL}, L_5, L_6,… After completing the switching process, the server will keep sending frames of the new bitstream. Similarly, the switching from L to H at frame n can be done by sending …, L_{n-1}, D_n^{LH}, H_{n+1}, … from the server.

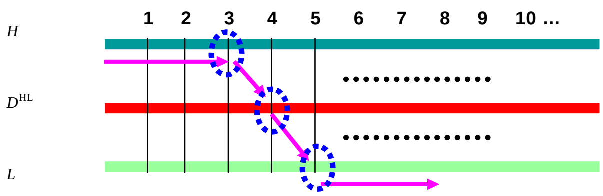


Fig. 3. An example of drift-compensated high-to-low-rate bitstream switching.

Table I shows the average PSNR comparison of drift-compensated and direct bitstream switching between 512 kbps and 1024 kbps. The two bitstreams are encoded with a frame rate of 30 fps and GOP size of 30 frames using an MPEG-4 encoder with TM5 rate control. The quantization parameter used for encoding the D^{HL} drift-compensation bitstream is 11. There are 3 switching operations (switching down, then switching up, and finally switching down again) in the experiments in Table I. The PSNR performance is evaluated by comparing the frames decoded from the drift-compensated and direct-switched bitstreams which are received at the client with that obtained from the drift-free (i.e., with perfect switching) sequence that is composed of corresponding frames directly decoded from the higher-rate (512 kbps) and lower-rate (1024 kbps) bitstreams.

Due to the temporal prediction used for encoding P- and B-frames, the drifting errors will propagate from the switch-point to the following frames until the next I frame is reached. This error propagation can lead to significant quality degradation if the drift is not well controlled.

Table I. PSNR performance of using drift-compensated and direct switching

sequence	PSNR with drift-compensated switching	PSNR with direct switching
Coastguard	35.68 dB	34.31 dB
News	34.89 dB	30.33 dB
Singer	34.81 dB	30.49 dB

The two bitstreams D^{HL} and D^{LH}, which are encoded based on the reconstructed images from H and L, can be used to compensate for the drifting errors caused by bitstreams switching. The effectiveness of drift reduction depends on the quantization parameters used for encoding the drift-compensation frames [4]. The finer the quantization step-size, the less the drift, but the higher the sending bit-rate and the storage cost. If D^{HL} and D^{LH} are losslessly encoded, there won't be any drift errors. Because the storage costs are much lower, and continuously decreasing nowadays, the increase on the storage cost for the drift-compensation bitstreams is not an important issue. Moreover, since D^{HL} and D^{LH} are all off-line encoded, the extra complexity required in the streaming video server is to decide which frames from which bitstreams should be transmitted when performing bitstream switching. This bitstream management cost is not high. The most attractive feature of the proposed method is that, like direct switching scheme, it just requires a standard video decoder to decode the switched bitsteams. Compared with the "seamless switching" scheme in [1] which involves a high-complexity non-standard decoder, the decoder complexity is reduced drastically, leading to a great cost reduction since the number of clients is usually high.

2.2. FGS video streaming with Drift-Compensated Bitstream Switching

Although FGS can support a wide range of bit-rates to ease the adaptation of channel variations, its coding efficiency is, however, significantly lower than the non-scalable coder [5]. The wider the bit-rate supported, the lower the coding efficiency. This drawback can be mitigated by Splitting the wide bit-rate range into a few narrower sub-regions and encoding multiple FGS bitstreams so that each sub-region is handled by one bitstream [1,2].

As shown in Fig. 4, there are two FGS bitstreams with different bit-rate base layers. The base layer of FGS bitstream 1 is encoded at a rate lower than that of FGS bitstream 2. If the bandwidth is out of the bit-rate range that FGS bitstream 1 can support, the client can switch to FGS bitstream 2 for better video quality. Since the FGS bitstreams' base layers are encoded at different bit-rates, drifting errors will occur when switching over the FGS bitstreams. Because the FGS enhancement layer does not participate in prediction and reconstruction, it has nothing to do with drifting errors. The mismatch of FGS base layer references is the major cause of the drifting errors when performing FGS bitstream switching. As mentioned above, we can compensate for the drifting errors by using drift-compensation bitstreams between these FGS base-layers.

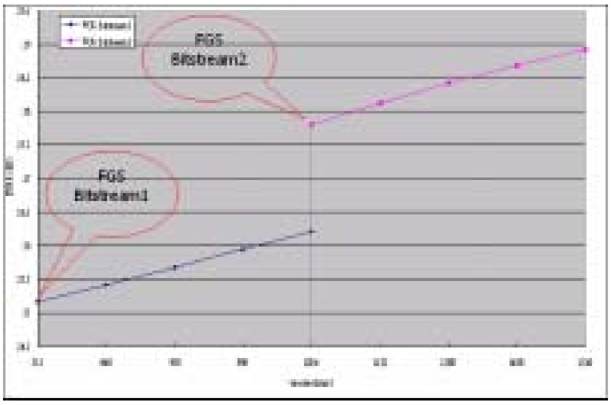


Fig. 4. Two FGS bit-streams with different bit-rates of base layer

3. RECEIVER-DRIVEN VIDEO MULTICASTING WITH DRIFT-COMPENSATED BITSTREAM SWITCHING

3.1. Receiver-Driven Rate Adaptation

In order to multicast FGS coded video bit-stream over heterogeneous networks, we should truncate the FGS enhancement layer into one or more sub-layers and transmit each to different multicast groups. In the case of truncating fixed bit-rates of enhancement layer into multicast groups, clients may receive unbalanced quality video in each frame. The advantage of separating one bit-plane into one group in the FGS enhancement layer is that clients can have an average visual quality of each frame. Besides, due to the scalable feature of FGS, we will not have the replicated problem of multiple bit-streams.

In our work, the server transmits one base layer and two or more enhancement sub-layers (MSB, MSB-1, etc.) to different multicast groups respectively. As a result, the clients can subscribe to one or more multicast groups with different levels of video quality according to their own capacity, bandwidth, and quality requirement.

In order to detect packet loss and calculate the packet loss ratio, we use the RTP (Real-time Transport Protocol) [9,10] on top of UDP to add the sequence number in the packet header. The clients keep tracking on the sequence number and frame number, to calculate the channel statistics and identify the lost frames and packets so as to calculate the packet loss ratio in an interval as follows:

$$P_k = \frac{L_k}{C_k} \tag{3}$$

where C_k is the interval with a constant number of packets and L_k is the number of lost packets in the interval C_k.

The clients can use the time-stamp of RTP to estimate *RTT*. First of all, assuming that the clocks of server and clients are well synchronized, the packets are time-stamped at the server side and sent to the Internet. The clients then receive these packets and report the receiving time. As a result, the *RTT* value can be estimated by subtracting the time-stamp of server from the receiving time of clients as described below.

$$RTT_i = 2 \times (t_i^c - t_i^s) \tag{4}$$

$$\overline{RTT}_k = \frac{1}{C_k} \sum_i^{C_k} RTT_i \tag{5}$$

where RTT_i denotes the round-trip time of packet *i*, t_i^c and t_i^s are respectively the time-stamps at the client and the server, and $\overline{RTT}_k$ denotes the average *RTT* in the interval C_k.

The TCP-friendly rate adaptation is necessary for multimedia applications to prevent congestion and unfair resource utilization in the Internet. Equation-based congestion control mechanisms [6,7] use the TCP throughput model [8] to adjust the transmission rate by estimating the TCP throughput R_{TCP} with the following equation.

$$R_{\mathrm{TCP}} = \frac{MTU}{RTT\sqrt{\frac{2p}{3}} + RTO\sqrt{\frac{27p}{8}}\,p(1+32p^2)} \tag{6}$$

where *MTU* is the maximum transmission unit, *RTO* is the retransmission time out, and *p* is the estimated packet loss rate.

3.2. Receiver-Driven Multiple FGS Bitstream Switching

In our system, before connecting to the server, the clients may configure their own connection such as cable modem (shared bandwidth), or ADSL (512 kbps ~ 6 Mbps). Then each client can decide which layers to subscribe to according to the initial set-up information. After receiving the video sequence for a period of time, clients may suffer from packet loss and/or insufficient bandwidth with their subscription level, i.e., these clients should leave some higher-level multicast groups they are subscribing to. For example, if one client subscribes to three multicast groups (one base layer and two enhancement layers) in the beginning and it turns out to detect packet loss by checking the sequence numbers of packets, this client will leave the highest-level multicast group to prevent packet loss. If the client still suffers from packet loss due to deficient bandwidth then it is also going to leave the second enhancement layer as illustrated in Fig. 5.

Now the client only subscribes to one multicast group (base layer), if packet loss occurs at this moment, the client cannot leave the latest multicast group, otherwise, the connection will be broken. Therefore, we add a multicast group with a lower bit-rate base layer to the system, and then clients are able to switch to the lower bit-rate layer from original base layer by leaving original multicast group to the lower one.

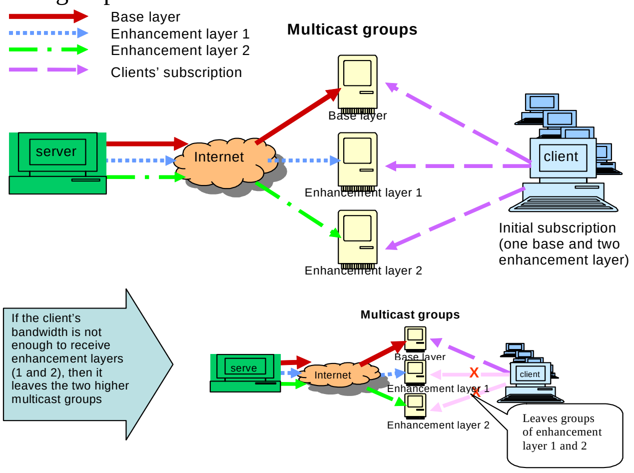


Fig. 5. Illustration of receiver-driven rate adaptation by leaving higher layer groups.

To avoid the drift caused by direct switching down (from higher-rate to lower-rate) or switching up (from lower-rate to higher-rate), we propose to add a drift-compensation bit-stream to reduce the drift errors. For example, if a client finds bandwidth is not enough according to the channel throughput estimation when receiving the third frame and the client must switch down. In order to solve the drift problem, we have to receive the fourth frame of the switching bit-stream and then the fifth, sixth, seventh frames of the lower-bit-rate bit-stream respectively. Fig. 6 illustrates the switching down example.

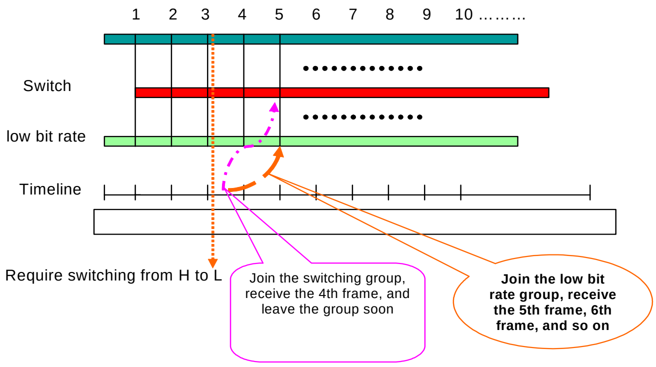


Fig. 6. Delayed switching bit-stream for multicasting

For video multicasting with multiple bitstream switching, we must ensure that the drift-compensation frames be received properly by the clients while switching. For example, as shown in Fig. 6, the 4th drift-compensation frame must be received before switching to the 5th frame of the lower-rate bitstream. Therefore, we propose to delay the drift-compensation frames by some time-slots in transmission so that the client can receive the correct video data to decode and display.

3.3. System Architecture

Fig. 7 shows the architecture of our client/server video streaming system. It allows multiple clients to connect to the server and stream the video programs from the server simultaneously. This system adopts FGS [5] as the coding tool. The FGS pre-encoded video bit-streams are stored in the server, and the clients must use an MPEG-4 FGS decoder for decoding. The network estimator plays an important role In the client part, it should estimate its bandwidth and capacity firstly. According to the available bandwidth estimated, the subscription adaptor will decide how to subscribe to multicast groups. Besides, the buffer manager is responsible for preventing from buffer underflow or overflow and has to cooperate with the subscription adaptor and the MPEG-4 decoder.

The server sends the base layer and enhancement sub-layers of the FGS coded video to different multicast groups. To support the multiple bit-stream switching mechanism, each video is encoded into two FGS bitstreams with distinct base-layer bit-rates, which are sent to two different multicast groups. The client chooses the proper one which matches its bandwidth the most. Two pre-encoded drift-compensation bitstreams are also sent to two other multicast groups so that clients can subscribe to one of them if bitstream switching (up or down) between the two FGS base-layers is required for rate adaptation. The number of enhancement sub-layers for each FGS bit-stream in our experiments is set to two, which are also sent to distinct multicast groups, respectively.

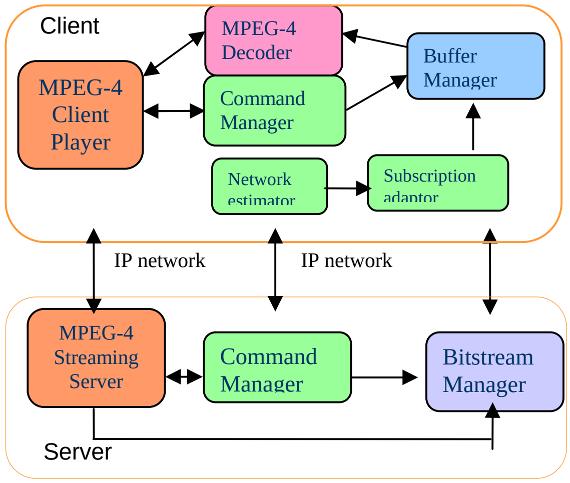


Fig. 7. Client/server architecture of the proposed video streaming system.

4. EXPERIMENTAL RESULTS

We encoded two CIF (352x288) video sequences "News" and "Singer," each with a GOP size of 14, at two bit-rates: 256 kbps and 128 kbps, respectively. The bit-rate of one base layer plus the MSB bit-plane of the enhancement layer is bout 300 kbps. The bit-rate of the base layer plus the first two enhancement sub-layers (MSB and MSB-1 bit-planes) is about 700 kbps. In the beginning, the clients subscribe to the higher-bit-rate base-layer and several enhancement sub-layers if their bandwidth is enough. Then the receiver-driven congestion control mechanism starts to estimate the network conditions, including the packet loss ratio, *RTT*, and compute the TCP throughput by Eq. (6). When a client observes the estimated available bandwidth is lower than its current subscription level, it will leave one higher enhancement sub-layer to avoid packet loss or congestion. Fig 8 shows the network variation model used in our experiments. We can observe from Fig. 8 that the subscription level adapts to the network conditions quite well according to the estimated TCP throughput.

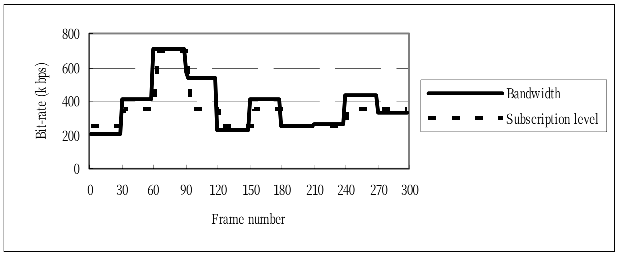
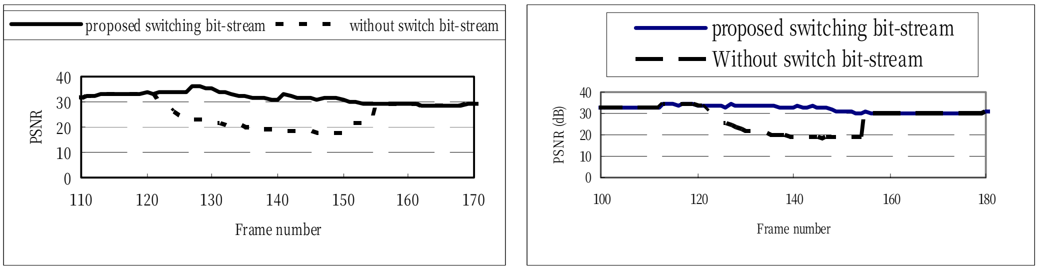


Fig. 8. Rate adaptation according to bandwidth estimation

By incorporating the drift-compensated switching algorithm presented in Section 3, the proposed system can further adapt to large bandwidth variations, while remaining reasonable visual quality. Furthermore, since our proposed system is capable of estimating the channel throughput, the clients can subscribe to the drift-compensation frames and the next bit-stream to switch to prior to switching when the estimated available bandwidth is close to the switching threshold. Since each frame is ensured to be prepared for decoding by grabbing in the buffer beforehand, the decoded video with the proposed method would be much smoother than the direct switching scheme.

In our experiments shown in Fig. 9, the channel bandwidth for one client is not enough and the client should switch to the lower-rate bitstream at frame #120. If the client leaves the higher-rate group and join the lower-rate one directly, drift errors will be introduced and propagate, thereby resulting in significant quality degradation as explained above. The quality degradation due to drift errors with the direct switching scheme can be observed in Fig. 9, while our proposed method can mitigate the drift drastically. In the experiments, only one switching event was triggered. The average PSNR improvement is about 1.5 dB. If the number of switching events is higher, the improvement will become more apparent.



(a) (b)

Fig. 9. PSNR comparison of proposed switching bit-stream and without switch bit-stream (a) "News" sequence (b) "Singer" sequence.

5. CONCLUSIONS

In this paper, we have proposed an MPEG-4 video streaming system with both unicast and multicast support. The proposed system can provide network adaptation with receiver-driven congestion control by estimating the network conditions and bandwidth capacity. In the proposed system. The values of *RTT*, packet loss ratio, and packet size were used to compute the TCP throughput. Using the information carried in the RTP (real-time transport protocol), the *RTT* and packet loss ratio can be estimated efficiently. With the estimated TCP throughput, clients are able to adjust their subscription levels by joining or leaving some multicast groups according to the estimated bandwidth. In addition to the congestion control mechanism, we also present a drift-compensated bitstream switching scheme to allow users to switch up and down over different-rate bitstreams with minimized drift errors. We also integrate the proposed switching scheme into our receiver-driven multicasting platform so that users can adapt to large network variations more efficiently. In the experiments, we showed that our proposed schemes can solve network congestion efficiently and achieve significant video quality improvement.

REFERENCES

[1] X. Sun, F. Wu, S. Li, and W. Gao, "The framework for seamless switching of scalable bitstreams," ISO/IEC JTC1/SC29/WG11/MPEG2002/m8214, Jeju Island, Mar. 2002.
[2] M. van der Scharr, "Using S-frames for fast switching between FGS streams and switching between MC-FGS structures to limit prediction-drift," ISO/IEC JTC1/SC29/WG11/MPEG2002/m8140, Jeju Island, Mar. 2002.
[3] M. Karczewic and R. Kurceren, "A proposal for SP-frames," ITU-T Q6/SG16, VCEG-L27.
[4] C.-W. Lin, J. Youn, J. Zhou, and M.-T. Sun, 201CMPEG video streaming with VCR-functionality," *IEEE Trans. Circuits Syst. Video Technol.* Vol. 11, pp. 415 -425, Mar. 2001.
[5] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 video standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 3 , pp. 301-317, Mar. 2001.
[6] S. Floyd, J. Pahdye, and J. Widmer, "Equation-based congestion control for unicast application," *SIGCOMM'00*, Aug. 2001.
[7] Y. Kim, J. Kim and C.-C. J. Kuo, "Smooth and fast rate adaptation mechanism (SFRAM) for TCP-friendly Internet video," in *Proc. Packet Video Workshop 2000,* Apr. 2000.
[8] J. Padhye, V. Firoiu and D. Towsley, "Modeling TCP throughput: a simple model and its empirical validation," *UMASS CMPSCI Technical Report, TR98-008,* Feb. 1998.
[9] Y. Kikuchi, T. Nomura, and S. Fukunaga, "RTP Payload Format for MPEG-4 Audio/Visual Streams," *Request for Comments (proposed standard) 3016. IETF,* Nov. 2000.
[10] S. Casner and V. Jacobson, "Compressing IP/UDP/RTP headers for low-speed serial links," *Request for Comments (proposed standard) 2508. IETF*, Feb. 1999.