

# ON THE RATE-DISTORTION PERFORMANCE OF DYNAMIC BITSTREAM SWITCHING MECHANISMS

Bo Xie

PacketVideo Corporation  
San Diego, CA 92121  
[xie@packetvideo.com](mailto:xie@packetvideo.com)

Wenjun Zeng

Dept. of Computer Science  
Univ. of Missouri-Columbia  
[zengw@missouri.edu](mailto:zengw@missouri.edu)

## ABSTRACT

*Bitstream switching is an effective way to deal with bandwidth variation in transmitting multimedia over time-varying channels. A number of bitstream switching mechanisms have been proposed to reduce the potential error drift introduced by bitstream switching, often without explicitly taking into account the impact of the switching on the bandwidth and delay requirements in their performance analysis. This paper studies the relative rate-distortion (R-D) performance of some of these bitstream streaming mechanisms by extending a delay-aware R-D optimized dynamic bitstream switching framework we previously proposed. By considering different switching mechanisms, we show that the proposed extended R-D based framework can adaptively choose the right switching mechanism and effectively switch at the right time to the right bitstream to achieve optimized rate-distortion performance.*

## 1 INTRODUCTION

Bit rate scalability is essential when transmitting video over time-varying channels such as the Internet or the wireless networks. However, a scalable bitstream alone may not provide a large enough scalable bit rate range to address large bandwidth variation without sacrificing the coding efficiency. In addition, current industrial standards, e.g. 3GPP, the Third Generation Partnership Project [1], may not necessarily support scalable codecs. As a result, dynamic bitstream switching among a set of bitstreams encoded at different bit rates may be the only way to effectively deal with bandwidth variation in a standard compliant way. The goals of dynamic bitstream switching are to achieve high utilization of the available bandwidth, and to avoid player rebuffering due to congestion, to produce the best content quality and experience for the end users.

A number of bitstream switching mechanisms have been proposed in the literature. One simple way is to switch only at I (Intra-coded) frames, at the cost of long delay in the switching or significant sacrifice of coding efficiency, depending on how frequently I frames are encoded. Recently, S/SP/SI frames have been proposed to serve as special bridging frames for switching from one bitstream to another [2][3][4]. These specially designed bridging frames can reduce or completely eliminate the error drifting effect often observed in bitstream switching, but are typically very complex. They typically require generating two intermediate bridging streams between any two bitstreams, each requiring a re-encoding process. The storage overhead for the bridging streams is also of concern. In addition, the bit overhead for transmitting a S/SP/SI frame can be as large as

transmitting an I-frame [3], which may have some negative impact in the bitstream switching scenario where the bandwidth is of the major concern.

In [5], we proposed *Smart P-frame Switching*, i.e., to switch from one bitstream to another at potentially any time instance, without generating any intermediate bitstreams. We proposed a source characteristics based fast switching algorithm using P-frame switching that can achieve well-controlled quality drift.

One common problem with all above mentioned solutions is that they only focus on the potential drifting artifact introduced by the switching algorithms, without necessarily considering when to switch and what the impact on the bandwidth and delay is. In other words, the rate-distortion (R-D) performance has not been analyzed sufficiently, and their relative R-D performance remains unclear.

In [6], we formulate a *delay-aware* rate-distortion optimized dynamic bitstream switching framework. By using an integrated generic end-to-end virtual network buffer management approach that explicitly takes into account the delay constraint, the proposed solution addresses both dynamic rate control and dynamic bitstream switching in a systematic way. The server makes the dynamic bitstream switching decisions by *proactively* estimating the end-to-end transmission delay, so as to avoid packet loss and player rebuffering. Only P-frame switching is considered as the switching mechanism in [6].

In this paper, we consider multiple candidates of switching mechanisms, and study the relative R-D performance of different bitstream streaming mechanisms by extending our delay-aware R-D optimized dynamic bitstream switching framework. We show that the proposed extended R-D based framework can adaptively choose the right switching mechanism and effectively switch at the right time to the right bitstream to achieve optimized rate-distortion performance.

## 2 AN OVERVIEW OF SOME RELATED WORK

We briefly review P-frame switching [5] and S-frame switching [2] in this section, as they will be used in our simulations. We also summarize the delay-aware R-D optimized dynamic bitstream switching framework we proposed in [6].

### 2.1 P-frame switching

P-frame switching simply sends the next encoded frame in the bitstream that it switches to, so it is the simplest implementation and is thus very attractive for live applications. It allows the server to perform a switch at any (I- or P-) frame, so switching

resolution is high, and there is no extra bit overhead. Standard bitstream syntax (from the decoder's point of view) is adhered to. The main concern with this method is that switching on P-frames introduces drift as the frame will be reconstructed by the client based on a reference frame that is in a different bitstream. Nevertheless, well-controlled quality drift can be achieved by intelligently choosing the best frame to switch within a small window [5].

## 2.2 S-frame switching

S-frame switching reduces the drift problem of P-frame switching through specially encoded frames. These frames are called "bridge frames" or S-frames. S-frame can be thought of as a normal frame (motion vectors + DCT) encoded using a reference frame from Stream A, but targeting some *already-coded* frame in Stream B. Transmitting the S-frame thus allows for a smooth transition from A to B. Some small drift still exists depending on how finely the S-frames are encoded. More importantly, as mentioned before, the bit overhead for transmitting a S frame can be as large as an I-frame size [3], which is a major concern in the bitstream switching scenario.

## 2.3 Delay-aware R-D optimized bitstream switching

Our work in [6] assumes a *virtual* network buffer located between the server and the client that abstracts the potentially complex network topology, and accounts for the delay and loss of packets, as shown in Fig. 1. Note that for streaming video over the best effort Internet/wireless networks, the intermediate network nodes' buffers may cause excessive delay. Denote  $B^n(i)$  and  $B^d(i)$ , respectively, as the virtual network buffer and the decoder buffer occupancies at the time instant  $i$  (when the  $i$ th frame is streamed). Let  $C(i)$  and  $R(i)$ , respectively, be the effective channel transmission rate observed/estimated at the receiver and the source rate (which is assumed to be equivalent to the server's sending rate) at the time instant  $i$ . Let us first assume there is no packet loss in the network. At any time instance  $i$ , we have

$$B^n(i) = \sum_{j=1}^i R(j) - \sum_{j=1}^i C(j) \quad (1)$$

$$B^d(i) = \begin{cases} \sum_{j=1}^i C(j) - \sum_{j=1}^{i-N} R(j) & \text{if } i \geq N \\ \sum_{j=1}^i C(j) & \text{otherwise} \end{cases} \quad (2)$$

where  $N$  is the number of frames that are accumulated in the decoder buffer before the decoder starts to decode and playback video. It accounts for the initial playback delay at the receiver.

Based on Eqs. (1) and (2), it turns out that the maximum allowable level of the virtual network buffer occupancy at the time instant  $i$  that ensures no decoder buffer underflow at the time instant  $i+N$  is

$$B^n(i)_{\max} = \sum_{j=i+1}^{i+N} C(j) \quad (3)$$

i.e., the condition for no decoder buffer underflow is

$$B^n(i) \leq \sum_{j=i+1}^{i+N} C(j) \quad (4)$$

Eq. (4) suggests that at any time  $i$ , the data in the virtual network buffer should be depleted within the next  $N$  frame time intervals. For best effort networks,  $C(j)$  is time varying and not directly controllable, although it can be estimated/predicted by the client. The allowable maximum virtual network buffer fullness is thus also time-varying. We assume the player has sufficiently large

decoder buffer size (with respect to the startup delay constraint) so decoder buffer overflow is less of a problem.

To ensure efficient network bandwidth usage, we should have

$$B^n(i) \geq 0 \quad (5)$$

Eqs. (4) and (5) together provide a generic virtual network buffer constraint for a streaming video application [6].

Based on this generic virtual network buffer constraint, a R-D optimized bitstream switching framework is formulated in [6] to achieve optimized R-D performance using P-frame switching.

## 3 ADAPTIVE SELECTION OF THE BITSTREAM SWITCHING MECHANISMS

In this section, we extend the delay-aware R-D optimized bitstream switching framework that deals with only single switching mechanism [6] to evaluate the relative effectiveness of multiple bitstream switching mechanisms in *on-demand* streaming applications.

For ease of presentation, let us assume there are only two candidate bitstreams to work with, and there are a set of candidate switching mechanisms that can be used, denoted as  $SM(k)$ ,  $k=1, \dots, M$ , where  $M$  is the total number of candidate switching mechanisms considered. For example, some of the candidates are P-frame switching [5], S-frame switching [2], and SP-frame switching [3][4], etc. The problem can be formulated as finding the best switching mechanism (among the candidates) and the optimal switching points that result in overall the least distortion, subject to the end-to-end virtual network buffer constraints as specified in Eqs. (4) and (5).

For bitstream switching between two bitstreams with source rates  $R_1(i)$  and  $R_2(i)$  ( $R_1(i) > R_2(i)$  on average) at time instant  $i$ , respectively, the problem can be formulated as follows. Assuming the effective channel bandwidth profile  $C(j)$  is known before hand ( $C(j)$  can only be estimated as shown later) and that the server is currently streaming Bitstream 1. At the time instant  $i$ , the server can choose to stay with Bitstream 1, or switch to Bitstream 2. If it chooses to stay with Bitstream 1, then the  $i$ th frame may be lost at the decoder, depending on the relationship between  $B^n(i)$  and  $B^n(i)_{\max}$ . The distortion observed at the decoder (for any  $SM(k)$  and any potential switching point  $l (> i)$ ) would be

$$D^{k,l}(i) = \begin{cases} D_1(i), & \text{if } B^n(i) < B^n(i)_{\max} \\ D_1(i) + D_1^{drift}(i), & \text{otherwise} \end{cases} \quad (6)$$

where  $D_1(i)$  is the distortion of the  $i$ th normal reconstructed frame in Bitstream 1, and  $D_1^{drift}(i)$  is the extra drifting distortion incurred when the previous decoded frame is displayed for the late packet.

If on the other hand, the server chooses to switch from Bitstream 1 to Bitstream 2 at the time  $l$  ( $l \leq i$ ) using the bitstream mechanism  $SM(k)$ , the distortion at the decoder would be

$$D^{k,l}(i) = \begin{cases} D_2(i) + D_2^{SM(k),l}(i), & \text{if } B^n(i) < B^n(i)_{\max} \\ D_2(i) + D_2^{SM(k),l}(i) + D_2^{drift}(i), & \text{otherwise} \end{cases} \quad (7)$$

where  $D_2(i)$  is the distortion of the  $i$ th normal reconstructed frame in Bitstream 2,  $D_2^{SM(k)l}(i)$  is the extra distortion caused by switching at time  $l$  using switch mechanism  $SM(k)$ ,  $D_2^{drift}(i)$  is the extra distortion caused by drifting when the previous decoded frame is displayed for the  $i$ th frame due to late packet.

For a potential switching point  $i$ , let  $D_k^{W,i} = \sum_{j \in W} D^{k,i}(j)$  be the accumulated distortion over a local common window of time  $W$ , e.g., a Real-Time Control Protocol (RTCP) [7] report period  $T_{RTCP}$  when real-time transport protocol (RTP) [7] is used, for the  $k^{\text{th}}$  candidate switching mechanism. The optimal switching point  $i^*$  and the best switching mechanism  $k^*$  are the ones that minimize the associated  $D_k^{W,i}$ , under the generic virtual network buffer constraints in Eqs. (4) and (5), i.e.,

$$(i^*, k^*) = \arg \min_{(i,k)} D_k^{W,i} = \arg \min_{(i,k)} \sum_{j \in W} D^{k,i}(j) \quad (8)$$

Note that for on-demand applications,  $D^{k,i}(j)$  or partial summation of  $D_k^{W,i}$  for each potential switching point  $i$  and each switching mechanism  $k$  can be pre-computed offline (by actually performing the switching) and stored as side information with the bitstreams to facilitate the on-the-fly decision making.

In the following, we will use the RTP streaming scenario to further illustrate how the proposed algorithm works. We use  $I$  to indicate the time instant a RTCP report is received.

$C(I)$  can be estimated as the effective throughput observed at the client in the past RTCP report period, which will be conveyed to the server through the client's RTCP report.  $C(I)$  is used as the channel rate in estimating the virtual network buffer occupancy at each frame instance in the next RTCP report period as well.

Upon receiving a RTCP report from the client, the server updates  $C(I)$  and the virtual network buffer occupancy:

$$B^n(I) = B^n(I-1) + (B_{sent} - B_{received}) \quad (9)$$

where  $B_{sent}$  and  $B_{received}$  are the number of bits sent by the server and received by the client, respectively, during the last RTCP report period. Note that  $B_{received}$  includes not only packets received by the client at the transport layer, but also *acknowledged/detected* lost packets that the client never receives at the transport layer, so that any known lost packets will be accounted/subtracted in the calculation of the virtual network buffer occupancy [6]. During the current RTCP report period,  $B^n(i)$  is estimated based on  $B^n(I)$  and Eq. (1). The server then performs the switching decision based on Eq. (8) subject to the virtual network buffer constraints of Eqs. (4) and (5).

## 4 EXPERIMENTAL RESULTS

Due to space limitation, we show only one example where we have two MPEG-4 bitstreams with bit rates of 128 kbps and 28 kbps, and frame rates of 15 fps and 3 fps, respectively. We only consider using two candidate switching mechanisms, i.e., the P-frame switching mechanism and the S-frame switching mechanism. We have an estimated bandwidth profile of the wireless network. In fact, the upper curve in Fig. 2 represents the

upper-bounds of the virtual network buffer occupancy, which is defined as a scaled version of the estimated instantaneous network bandwidth. So the upper curve in Fig. 2 essentially shows the network bandwidth fluctuation. At the beginning of the streaming, the bitstream with a bit rate of 128 kbps is chosen. This bit rate is larger than the instantaneous network bandwidth, so the estimated virtual network buffer occupancy gets accumulated, until the time instance of about 11s where the virtual network buffer occupancy upper bound is about to be reached. The proposed algorithm then chooses to switch to the other bitstream with lower bit rate at a point/frame that is R-D optimized. In fact, it chooses to use P-frame switching, again based on the R-D criteria. At about 15s, the network bandwidth increases. The algorithm then decides to switch back to the bitstream with higher rate to avoid virtual network buffer underflow, this time by choosing to use S-frame switching against P-frame switching.

Fig. 3 shows the PSNR curves of various switching algorithms, including a heuristic client-based approach that switches immediately simply upon a request from the client based on the client buffer fullness [6]. Figs. 3 shows that the proposed delay-bounded R-D based approach has a quick response to the bandwidth change. It can achieve a nice bitstream transition without introducing significant annoying visual artifact, by choosing the right switching mechanism and switching point.

For comparison, Fig. 3 also shows the PSNR curve of the delay-bounded R-D based approach that uses only P-frame switching, as was used in [6]. We make some observations in the following.

- 1) When switching from low rate bitstream to high rate bitstream, S-frame switching is usually preferred. Let us look at the data from 15s to 20s in Fig. 3. The actual switching point in the low rate bitstream is almost the same for both switching mechanisms. Apparently, S-frame switching reduces the quality drift a lot and thus becomes the better choice in this case.

In many cases, when switching from low rate bitstream to high rate bitstream, the best switching strategy is to do S-frame switching as early as possible to take advantage of as many high quality frames as possible. In some other cases when the channel rate does not increase much or the high source rate is still large compared to the increased channel rate, S-frame switching may not necessarily be the winner. In this case, there will be a competition between earlier P-frame switching (earlier S-frame switching would cause virtual network buffer overflow) and later S-frame switching (most likely one frame later). For the earlier P-frame switching, at least one more potentially higher quality/rate frame is obtained, but with a penalty, i.e., the typically larger quality drift. For the later S-frame switching, there is at least one more potentially low quality/rate frame but the quality drift is reduced. Therefore, there would be an R-D tradeoff between earlier P-frame switching and later S-frame switching. In most cases, quality drift of P-frame switching is larger than quality difference between one high rate frame and the corresponding low rate frame. Then S-frame is preferred.

2) When switching from high rate bitstream to low rate bitstream, the result shows that we often end up with P-frame switching. The reason is that, in this case, staying with high rate bitstream will inevitably cause virtual network buffer overflow as the maximum allowable virtual network buffer occupancy (see Eq. (3)) decreases due to reduced channel rate. Our R-D criterion suggests switching as late as possible in this case. Therefore, there exists a point/frame (say Frame  $K$ ) from which P-frame switching does not cause virtual network buffer overflow, but P-frame switching at the next frame causes virtual network buffer overflow. Now consider the S-frame case. S-frame switching at Frame  $K$  will usually cause virtual network buffer overflow because the S-frame size is usually much larger than the frame size of Frame  $K+1$  and P-frame switching at Frame  $K+1$  causes virtual network buffer overflow. So most likely S-frame switching at earlier than Frame  $K$  (say, Frame  $K-1$ ) will avoid causing virtual network buffer overflow. Here comes the competition between earlier S-frame switching and later P-frame switching. Earlier S-frame switching may gain from quality drift reduction but incur the quality loss from (at least) one more low quality/rate frame. In this scenario, the quality drift due to P-frame switching may not be larger than the quality difference between a high quality/rate frame and the corresponding low quality/rate frame. The reason is that when reconstructing the “switch-to” frame, the original low quality reference frame in the new bitstream is replaced by a high quality reference frame in the previous high quality bitstream. In fact, in the experiments, we observe that in many cases of switching from high rate to low rate, the quality drift due to P-frame switching is even smaller than the quality drift due to S-frame switching (with QP=3), in which cases we end up with P-frame switching. These cases often occur when switching to a frame that has truly time-wise aligned referenced frames in the two bitstreams.

In summary, the virtual network buffer constraints are more effective when switching from high rate to low rate. In this case, virtual network buffer overflow becomes critical in choosing the switching mechanism. Usually quality drift reduction does not compensate for the quality difference between a high quality frame and the corresponding low quality frame. In such cases, P-frame switching is preferred. When switching from low rate to high rate, the maximal allowable virtual network buffer occupancy increases, and can often easily accommodate an extra S-frame for the gain of quality drift reduction. Furthermore, S-frame switching does not necessarily reduce the quality drift when compared to P-frame switching, especially in switching from high rate bitstream to low rate bitstream.

## 5 CONCLUSION

This paper studies the relative R-D performance of different bitstream streaming mechanisms by extending our delay-aware R-D optimized dynamic bitstream switching framework. The proposed extended R-D based framework can adaptively choose the right switching mechanism and effectively switch at the right time to achieve optimized rate-distortion performance. Work currently under investigation includes considering switching

among multiple bitstreams and including other switching mechanisms in the framework.

*Acknowledgement:* This work was supported, in part, by a grant from the University of Missouri System Research Board, and in part, by NSF grant CNS-0423386.

## 6 REFERENCES

- [1] 3GPP TS 26.234, v 1.5.1 (2001-03), 3<sup>rd</sup> Generation Partnership Project, TSG-SA, “Transparent end-to-end packet switched streaming service, Protocols and codecs, Release 4”
- [2] N. Faerber and B. Girod, “Robust H.263 compatible video transmission for mobile access to video servers”, *Proc. IEEE Inter. Confer. Image Processing*, 1997.
- [3] M. Karczewicz and R. Kurceren, “A proposal for SP-frames”, document VCEG-L-27, ITU-T Video Coding Experts Group Meeting, Eibsee, Germany, 09-12 January 2001
- [4] R. Kurceren and M. Karczewicz, “Improved SP-frame encoding”, document VCEG-M-73, ITU-T Video Coding Experts Group Meeting, Austin, TX, 02-04 April 2001
- [5] B. Xie and W. Zeng, “Source characteristics based fast bitstream switching,” in *IEEE Inter. Conf. Multimedia & Expo*, July 2003.
- [6] B. Xie and W. Zeng, “Rate-distortion optimized dynamic bitstream switching for scalable video streaming,” in *IEEE Inter. Conf. Multimedia & Expo*, June 2004.
- [7] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A transport protocol for real-time applications,” IETF RFC 1889, January 1996.

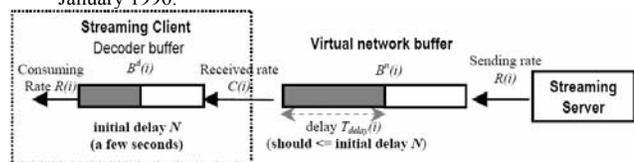


Fig. 1: A generic virtual network buffer model.

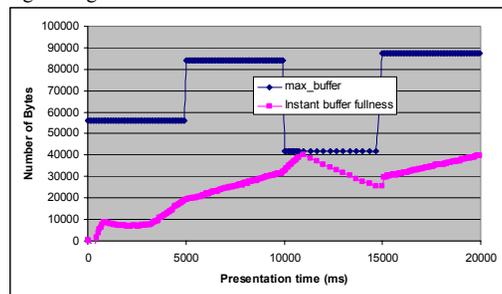


Fig. 2: Virtual network buffer fullness vs upperbound.

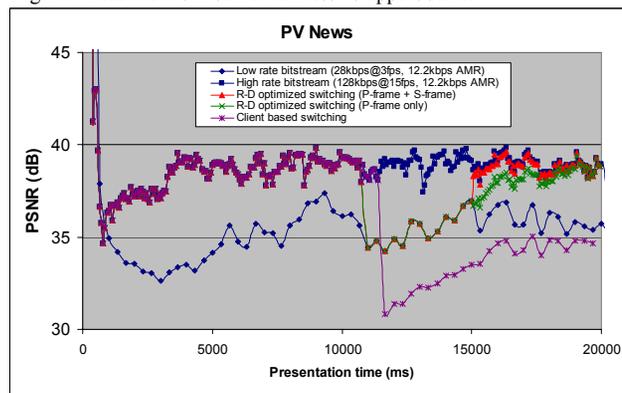


Fig. 4: PSNRs for different bitstream switching strategies.