REDUNDANT REPRESENTATION FOR NETWORK VIDEO STREAMING USING RECONSTRUCTED P-FRAMES AND SP-FRAMES

Gene Cheung

National Institute of Informatics, Tokyo, Japan

Wai-tian Tan

Hewlett-Packard Laboratories, Palo Alto

ABSTRACT

For low-delay streaming of pre-encoded video over lossy networks, fast recovery from decoding errors typically involves use of frequent intra-coded frames, which incurs high bandwidth cost. In this paper, we present a redundant representation of video using a bandwidth-efficient but non-resilient main bitstream, and an additional auxiliary bitstream dedicated to recovery from losses in the main bitstream. In particular, we insert primary SP-frames periodically into the main bitstream, and encode corresponding secondary SP-frames and reconstructed P-frames in the auxiliary stream. When a frame loss occurs, a reconstructed P-frame is first sent to re-synchronize decoder back to normal motion compensation loop, then a secondary SP-frame corresponding to the location of the next pre-inserted primary SP-frame in the main stream is sent thereafter, eliminating coding drift. Results show that proposed method out-performs non-redundant representation of I-frame insertion by up to 11 frames in recovery time, and out-performed redundant representation of only reconstructed P-frames by up to $2.2dB$ in average PSNR.

***Index Terms*—** Video streaming, SP-frames, redundant representation

1. INTRODUCTION

In streaming of video over lossy networks that requires low delay, a necessarily small playback buffer at the client's decoder translates to a limited number of timely server-to-client retransmissions, meaning that occasionally some frames may not be successfully recovered in time for playback. The resulting decoding errors—distortion from undecodeability of irrecoverably lost frames and subsequent frames differentially encoded using as predictors these lost frames—persist till the next synchronization frame (e.g., intra-coded I-frame) transmitted in the stream, creating a potentially long period of poor quality for the viewer. For live-encoded video such as conversational video, this decoding errors can be effectively contained using a scheme called *newpred*, that adaptively encodes new frames using known successfully decoded frames in the near past for prediction. However, for low-delay streaming of pre-encoded video, important for applications such as interactive network video browsing, the problem is more difficult.

One approach to the problem is to simply increase the I-frame insertion frequency in the bitstream to contain frame losses. Because an I-frame is several times larger than a P-frame, this approach increases the streaming bitrate even when there are no losses, resulting in bandwidth inefficiency. A better alternative is to employ a *redundant representation*, where an original picture is pre-encoded into *multiple* frames, one each for a *main bitstream* and an *auxiliary bitstream* prior to streaming, as shown in Fig. 1. More specifically, an efficiently compressed but non-loss-resilient *main bitstream* is transmitted when no frame losses are experienced. An *auxiliary bitstream* that is designed for quick recovery is deployed during frame losses. After decoding errors are contained, regular streaming is resumed using the main bitstream.

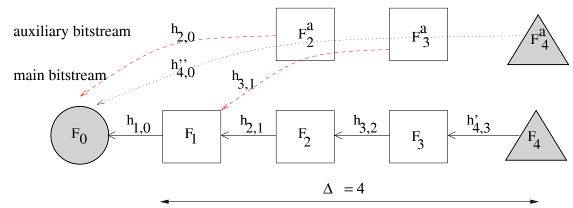


Fig. 1. Example of Main and Auxiliary Bitstreams

In this paper, we propose to construct the redundant representation as follows. We first pre-insert *primary SP-frames* [1] (triangles in Fig.1), a new frame type in H.264 coding standard [2], periodically into the main bitstream. Into the auxiliary bitstream, we then pre-encode *reconstructed P-frames* (one for each loss event) and *secondary SP-frames* (at same frame indices as the primary SP-frames in the main bitstream). When a frame F_i in the main bitstream is irrecoverably lost at decoder, a reconstructed P-frame $F^a_{\alpha_i}$ in the auxiliary bitstream corresponding to loss of F_i, using F_{i-1} as predictor, is sent to re-synchronize the decoder back to the normal motion compensation loop. Frames in the main bitstream F_j's, $j > \alpha_i$, can then be sent to the client without alteration for decoding.

Because reconstructed P-frame $F^a_{\alpha_i}$ in the auxiliary bitstream is not exactly the same as F_{α_i} used in the motion compensation loop during main bitstream encoding, sending subsequent F_j's, $j > \alpha_i$, in the main bitstream using $F^a_{\alpha_i}$ as predictor will cause a coding drift [3]. Hence before the drift grows to a sizable distortion, we send a secondary SP-frame, corresponding to the location of the next primary SP-frame in the main bitstream, to eliminate it (secondary SP-frame is reconstructed to be exactly the same as primary SP-frame).

In contrast to the non-redundant representation approach previously mentioned that simply increases I-frame insertion frequency, our redundant approach prepares an auxiliary bitstream for error recovery using extra storage, *without* consuming more bandwidth during loss-free duration. Auxiliary bitstream encodes the same number of frames as the main bitstream, resulting in roughly twice the storage required; given the rapid decrease in storage cost per byte in commercial market, this is a sensible tradeoff. Results under typical burst-loss network conditions show that our proposed redundant representation using reconstructed P-frames and SP-frames out-performed non-redundant representation of I-frame insertion in loss recovery time by up to 11 frames, and out-performed redundant representation using reconstructed P-frames only by up to $2.2dB$.

The outline of the paper is as follows. We first discuss related work in Section 2. We then present source and network models used in Section 3. The crux of our proposed redundant representation is the determination of the optimal primary SP-frame period in the main bitstream; we present the chosen objective function and efficient methods to calculate it in Section 4. We present an offline algorithm to find the optimal primary SP-frame insertion period in Section 5. Finally, we present results in Section 6.

978-1-4244-4296-6/10/$25.00 ©2010 IEEE

2. RELATED WORK

While video streaming over burst-loss networks has been studied [4, 5], our work differs in that low latency is required, and we use data redundancy in video representation to contain error propagation using reconstructed P-frames and SP-frames during loss events.

Coding drift [3] is a well known coding phenomenon; SP-frames in H.264 [1] was designed to eliminate such drift when switching among different video streams (e.g., streams of the same video but of different bitrates) without resorting to the bandwidth-expensive intra-coded I-frame. Our proposal is unique in that drift is introduced on purpose in a controlled manner using reconstructed P-frames and is then subsequently eliminated using secondary SP-frames.

Redundant representation for video, where each original picture is pre-encoded into multiple frames, has been proposed for interactive multiview video streaming [6], where, as an observer interactively selects views as video is played back in time, a suitable differentially encoded version of a picture is selected at server for transmission given the observer's traversal of views to date. In contrast, this work is an application of redundant representation for video streaming over burst-loss networks.

3. SOURCE & NETWORK MODELS

3.1. Source Model

We assume frame rate of f frames per second is maintained during playback. We model each frame i, F_i, as a node in a directed acyclic graph (DAG) as shown in Fig. 1. Associated with each frame F_i is a deadline (not shown), upon which F_i must be delivered to the client for decoding and display or it will be rendered useless. A solid arrow $E(i, i-1)$ in Fig. 1 from F_i to F_{i-1} indicates that in the main bitstream a P-frame (square) or primary SP-frame (triangle) F_i is motion-compensated using the previous frame F_{i-1} as predictor, resulting in $h_{i,i-1}$ or $h'_{i,i-1}$ packets. Only the first frame in the sequence is encoded as I-frame (circle), and each subsequent primary SP-frame is located at multiples of SP-frame period Δ, for a total of M primary SP-frames. Fig. 1 shows an example when $\Delta = 4$.

During a streaming session, if a frame F_i is irrecoverably lost during transmission, a reconstructed P-frame $F^a_{\alpha_i}$, referencing F_{i-1} and pre-encoded in the auxiliary bitstream, can be transmitted to re-synchronize the decoder back to normal motion compensation loop. We model the overhead of each reconstructed P-frame $F^a_{\alpha_i}$ using $E(\alpha_i, i-1)$, a red dashed arrow in Fig. 1, each with associated size $h_{\alpha_i,i-1}$ in packets. Because $F^a_{\alpha_i}$ is not exactly the same as the original P-frame F_{α_i} in the main bitstream, there will be a *coding drift*; we model the resulting coding drift error in client's decoded frame F_k, $k > \alpha_i$, stemming from $F^a_{\alpha_i}$ as $\theta(\alpha_i, k)$.

To eliminate drift caused by a reconstructed P-frame $F^a_{\alpha_i}$, a subsequent secondary SP-frame $F^a_{\delta_i}$ is sent at one of primary SP-frame locations, $\delta_i = m\Delta$, $m \in \mathcal{I}^+$. Secondary SP-frame eliminates drift since by definition [1] it can be reconstructed exactly the same as the stored primary SP-frame in the main bitstream. We model the overhead of secondary SP-frames using blue dotted arrows, $E(\delta_i, i-1)$ in Fig. 1, each with associated size $h''_{\delta_i,i-1}$ in packets.

As an example, in Fig. 1 after losing F_1, server may transmit reconstructed P-frame F^a_2 ($\alpha_1 = 2$) referencing F_0, send pre-encoded F_3, then send secondary SP-frame F^a_4 ($\delta_1 = 4$) referencing F_0.

Finally, if no correctly decodeable frame of any kind arrives in time for F_k, then the most recently correctly decoded frame F_i is used as a replacement, with a resulting distortion $\Theta(i, k)$.

3.2. Network Model

We assume a network with constant bandwidth of C kbps and a Gilbert packet loss process with parameters p and q, shown in Fig. 2.

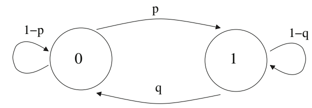


Fig. 2. Gilbert model for packet losses with parameters p and q.

p and q correspond to the state transition probabilities, and loss rate and average burst length are $\pi = p/(p+q)$ and $1/q$, respectively.

3.3. Some Useful Definitions

Given Gilbert model of parameters p and q, several useful quantities can be computed as done in [4] and others. For $i \geq 0$, we denote by $P(i)$ the probability of having at least i consecutive delivered packets following a lost packet, and by $p(i)$ the probability of having *exactly* i consecutive delivered packets between two lost packets:

$$p(i) = \begin{cases} 1-q & \text{if } i=0 \\ q(1-p)^{i-1}p & \text{otherwise} \end{cases}$$

$$P(i) = \begin{cases} 1 & \text{if } i=0 \\ q(1-p)^{i-1} & \text{otherwise} \end{cases}$$

Similar terms $q(i)$ and $Q(i)$ are defined by reversing the role of q and p. The probability of *exactly* m losses in n packets after an observed lost packet, $R(m, n)$, is given by:

$$R(m,n) = \begin{cases} P(n) & \text{for } m=0 \text{ and } n \geq 0 \\ \sum_{i=0}^{n-m} p(i) R(m-1, n-i-1) & \text{for } 1 \leq m \leq n \end{cases}$$

The probability of *exactly* m losses in n packets between two lost packets after an observed lost packet, $r(m, n)$, is given by:

$$r(m,n) = \begin{cases} p(n) & \text{for } m=0 \text{ and } n \geq 0 \\ \sum_{i=0}^{n-m} p(i) r(m-1, n-i-1) & \text{for } 1 \leq m \leq n \end{cases}$$

The probability of *exactly* m losses in n packets after a lost packet and preceding a received packet, $\bar{r}(m, n)$, is:

$$\bar{r}(m,n) = R(m,n) - r(m,n)$$

We define the complementary function $S(m, n)$ as the probability of having *exactly* m correctly received packets in n packets following an observed correctly received packet. $s(m, n)$ and $\bar{s}(m, n)$ are similarly defined counterparts to $r(m, n)$ and $\bar{r}(m, n)$. $S(m, n)$, $s(m, n)$ and $\bar{s}(m, n)$ are written similarly as $R(m, n)$, $r(m, n)$ and $\bar{r}(m, n)$ with $Q(n)$ and $q(n)$ in place of $P(n)$ and $p(n)$.

4. PROBLEM DEFINITION

We investigate the problem of finding the optimal primary SP-frame insertion period for server-client streaming of stored video. Intuitively, if the size of a primary SP-frame is no larger than a P-frame, then one would insert it as frequently as possible, so that coding drift can be eliminated with the transmission of a corresponding secondary SP-frame whenever one deems necessary. In practice, however, primary SP-frame can be a fair bit larger than a P-frame, and so inserting primary SP-frame too frequently would create bandwidth inefficiency, resulting in higher probability of frame decoding errors at the client if bandwidth is scarce.

4.1. Defining Objective Function

To investigate the appropriate primary SP-frame insertions, we define an objective function focusing on the minimization of two opposing adverse effects at the client: i) frame-level decoding error (pre-insertion of larger primary SP-frames leads to more irrecoverable frame losses); and ii) coding drift (starting from a reconstructed

P-frame sent following the first irrecoverable frame loss, till a secondary SP-frame sent to eliminate drift). The optimal primary SP-frame strategy is one that minimizes the sum of these two effects.

Let D be the expected distortion in a *steady-state Δ-frame sub-sequence* at the streaming client. By steady-state, we mean a representative frame series with $\Delta - 1$ P-frames and a primary SP-frame repeating infinitely. By analyzing the expected distortion of steady-state sub-sequences of different lengths Δ's, we can select the optimal primary SP-frame insertion period Δ^*.

For simplicity, we assume a server performs the following transmission strategy. Frames are transmitted in order of playback deadlines given a finite number of *transmission opportunities*, dictated by channel bandwidth C. Assuming further that the loss status of each transmitted packet is known instantly at the server, each lost packet is retransmitted until transmission success or until transmission opportunities have been depleted, the latter of which signals an irrecoverably lost packet of a frame. A frame is subsequently declared lost if a packet of the frame is irrecoverably lost.

Let $L_f(i)$ be the *first-loss probability* that frame F_i is the first irrecoverably lost frame in the Δ-frame sub-sequence. Let $L_g(i)$ be the *general-loss probability* that F_i is irrecoverably lost in general (first or subsequent loss). We can write D as:

$$
\begin{aligned}
D &\approx \sum_{i=1}^{\Delta} L_f(i) d_f(i, \alpha_i, \delta_i) + \frac{1}{\Delta} L_g(i) d_g(i, \alpha_i, \delta_i) \\
d_f(i, \alpha_i, \delta_i) &= \sum_{k=\alpha_i}^{\delta_i - 1} \theta(\alpha_i, k) \\
d_g(i, \alpha_i, \delta_i) &= L_r(i, \alpha_i) \sum_{k-i}^{\alpha_i - 1} \Theta(i-1, k) + \\
& \quad (1 - L_r(i, \alpha_i)) \sum_{k=i}^{\alpha_{\alpha_i}} \Theta(i-1, k) \qquad (1)
\end{aligned}
$$

where $d_f(i, \alpha_i, \delta_i)$ is the resulting drift of using reconstructed P-frame $F^a_{\alpha_i}$ and secondary SP-frame $F^a_{\delta_i}$ given first-lost frame F_i, and $d_g(i, \alpha_i, \delta_i)$ is resulting expected distortion given general-lost frame F_i. $d_g(i, \alpha_i, \delta_i)$ itself is a two-part penalty:

1. Undecodeable distortions $\Theta(i-1, k)$'s from lost frame F_i till reconstructed P-frame $F^a_{\alpha_i}$ given $F^a_{\alpha_i}$ is delivered successfully with *recovery probability* $L_r(i, \alpha_i)$.
2. Undecodeable distortions $\Theta(i-1, k)$'s from lost frame F_i till the following reconstructed P-frame $F^a_{\alpha_{\alpha_i}}$ if $F^a_{\alpha_i}$ is also lost.

4.2. Computation of Loss and Recovery Probabilities

We now discuss methods to compute first-loss probability $L_f(i)$, general-loss probability $L_g(i)$, and recovery probability $L_r(i, \alpha_i)$ in order to calculate distortion D. Central to the calculation of these probabilities is the discrete-time, discrete-valued stochastic process $\omega_i, i = 1, \ldots, \Delta$: the number of available packet transmission opportunities for frame F_i. We track how ω_i evolves as discrete time i progresses using a *discrete time Markov chain* [7]. Each column i in a Markov chain represents the available transmission opportunities ω_i for frame F_i. $\omega_i = 0$ denotes the state where F_i is irrecoverably lost. The Markov chain for a 4-frame sub-sequence is illustrated in Fig. 3. Each state $\omega_i = x$ of frame F_i, $1 \leq i < \Delta$, can transition to a state $\omega_{i+1} = y$ of the next frame F_{i+1} (black solid arrows), or transition to the failure state of F_i (red dashed arrows) with transition probabilities $Z(i, x; i+1, y)$ and $Z(i, x; i, 0)$, respectively. States $\omega_\Delta = x, \forall x$, of the last frame F_Δ will transition to states $\omega_1 = y, \forall y$, in the first frame F_1 in steady state.

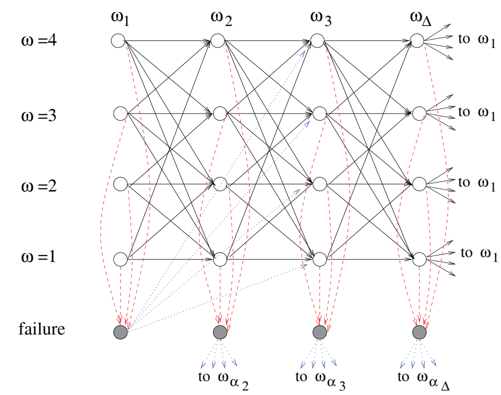


Fig. 3. Example of Transmission Opportunity Markov Chain

At failure state $\omega_i = 0$, a recovery attempt using reconstructed P-frame $F^a_{\alpha_i}$ is made. If successful, one transitions to steady states $\omega_{\alpha_i+1} = y, \forall y$, at frame F_{α_i+1} (blue dotted arrows) with probabilities $Z(i, 0; \alpha_i + 1, y), \forall y$. We discuss how these transition probabilities are calculated next.

We first note that when switching from a state $\omega_i = x$ to a state of the next frame F_{i+1}, ω_{i+1} receives a replenishment of one frame interval worth of transmission opportunities due to the later playback deadline of F_{i+1} relative to F_i. This is denoted by $g(1/f)$, where *opportunity generating function* $g(\tau)$ given available time τ is defined as:

$$g(\tau) = \lfloor \tau \times C / s_{pkt} \rfloor \qquad (2)$$

Basically $g(\tau)$ returns the maximum number of packets that can be delivered from server to client given transmission time τ.

For ω_{i+1} of frame F_{i+1} to take on value $y + g(1/f)$ starting at state $\omega_i = x$ of frame F_i, F_i must have exhausted $x - y$ opportunities for successful delivery of $h_{i,i-1}$ packets. This happens if F_i used exactly $x - y - 1$ opportunities to deliver the first $h_{i,i-1} - 1$ packets, with the immediately following packet transmitted successfully. Hence we can write $Z(i, x; i+1, y + g(1/f))$ as[1]:

$$Z(i,\ x;\ i+1,\ y + g(1/f)) = s(h_{i,i-1} - 1,\ x - y - 1) \qquad (3)$$

The computation of transition probability $Z(i, x; i, 0)$ depends on the number of opportunities x relative to the number of packets $h_{i,i-1}$ of F_i: if $x < h_{i,i-1}$, then F_i cannot possibly be transmitted successfully. If $x \geq h_{i,i-1}$, then F_i fails if fewer than $h_{i,i-1}$ packets were successfully transmitted given x attempts:

$$Z(i, x; i, 0) = \begin{cases} 1 & \text{if } x < h_{i,i-1} \\ \sum_{k=0}^{h_{i,i-1}-1} S(k, x) & \text{o.w.} \end{cases} \qquad (4)$$

Finally, we compute transition probability $Z(i, 0; \alpha_i + 1, y)$ from a failure state $\omega_i = 0$ of F_i to states $\omega_{\alpha_i+1} = y, \forall y$, of F_{α_i+1}: $Z(i, 0; \alpha_i + 1, y + g(1/f))$ is the probability that reconstructed P-frame F_{α_i} is delivered successfully with y transmission opportunities left unused, given $A = g((\alpha_i - i)/f)$ available opportunities:

$$
\begin{aligned}
Z(i,\ 0;\ \alpha_i + 1,\ y + g(1/f)) &= (1 - \pi)\, s(h_{\alpha_i, i-1} - 1,\ A - y - 1)) \\
& \quad + \pi\, \bar{r}(A - y - h_{\alpha_i, i-1},\ A - y - 1)) \\
A &= g((\alpha_i - i)/f) \qquad (5)
\end{aligned}
$$

[1]We assume here the channel is in good state initially—hence the use of $s(.,.)$ instead of $r(.,.)$—since transmission of F_{i+1} follows successful transmission of the last packet of previous frame.

We normalize the transition probabilities out of state $\omega_i = 0$ by $W_i = 1/\sum_y Z(i, 0; \alpha_i + 1, y)$, so that the sum of transition probabilities out of the state is one.

Assuming that the Markov chain constructed above is irreducible and aperiodic, there exists a unique *invariant probability vector* $\bar{v}$ for *transmission matrix* $\mathbf{Z}$ of the chain using transmission probabilities $Z(i, x; j, y)$'s discussed above, according to the Perron-Frobenius Theorem [7]; i.e.,

$$\bar{v}\,\mathbf{Z} = \bar{v}, \qquad \lim_{n\rightarrow\infty} \bar{\phi}\,\mathbf{Z}^n = \bar{v} \tag{6}$$

In other words, $\bar{v}$ is the eigenvector associated with the eigenvalue 1 for matrix $\mathbf{Z}$. Hence $\bar{v}$ can be found via eigen-decomposition, or alternatively by applying an initial distribution vector $\bar{\phi}$ to $\mathbf{Z}$ and letting the Markov chain transition till steady state.

4.2.1. *First-loss Probability*

Having derived probability vector $\bar{v}$, we approximate first-loss probability $L_f(i)$ as the likelihood that frame F_i encounters decoding error during transmission relative to other frames:

$$L_f(i) \approx \frac{P(\omega_i = 0)}{\sum_j P(\omega_j = 0)} \tag{7}$$

This is an approximation since $P(\omega_i = 0)$ encompasses both first time and subsequent decoding error of F_i during transmission.

4.2.2. *General-loss Probability*

Given $\bar{v}$, where each element in $\bar{v}$ represents the fraction of time spent at state $\omega_i = x$, general-loss probability $L_g(i)$ is simply a weighted sum of transition probabilities $Z(i, x; i, 0)$'s:

$$L_g(i) = \sum_x \left[\frac{P(\omega_i = x)}{\sum_{y>0} P(\omega_i = y)} \right] Z(i, x; i, 0) \tag{8}$$

4.2.3. *Recovery Probability*

For recovery probability $L_r(i, \alpha_i)$, it is simply the sum of probabilities out of state $\omega_i = 0$; i.e.,

$$L_r(i, \alpha_i, \delta_i) = \sum_y Z(i, 0; \alpha_i + 1, y) = 1/W_i \tag{9}$$

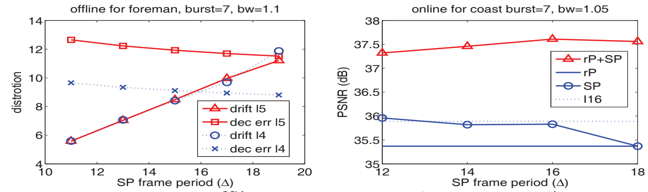
5. OFFLINE ALGORITHM

Having defined the objective function, we locally search for the best primary SP-frame insertions as follows. For a given primary SP-frame insertion period Δ, after the first I-frame we encoded the proceeding $\Delta - 1$ frames as P-frame and the following frame as primary SP-frame to construct a Δ-frame sub-sequence. For each frame F_i in the sub-sequence, we locally search for the optimal reconstructed P-frame $F^a_{\alpha_i}$ and secondary SP-frame $F^a_{\delta_i}$. We then construct the transmission matrix $\mathbf{Z}$ and find the invariant probability $\bar{v}$. First-loss probabilities $L_f(i)$'s, general-loss probabilities $L_g(i)$'s, and recovery probabilities $L_r(i, \alpha_i, \delta_i)$'s are then computed, which are subsequently used to calculate distortion D using (1).

The procedure is then repeated for all possible values of SP-frame period Δ, and the one with the smallest distortion is designated as the optimal SP-frame insertion period.

6. EXPERIMENTATION

We used JM H.264 version 14.2 to encode CIF size (352×288) sequences `foreman` and `coast`, using quantization parameter 20. For offline optimization, we set Gilbert parameters for average burst length of 7, and average loss rates of 0.05 and 0.04 in two independent trials. Available bandwidth C was set to $1.1\times$ the average bitrate of all I and P-frames only.



(a) `foreman` offline (b) `coast` online

Fig. 4. Simulation Results for Offline and Online Optimization

We plotted the calculated expected drift and decoding error distortions in our offline optimization as a function of Δ for `foreman` in Fig. 4(a) for two loss rates. We observe that as Δ increased, drift distortion increased due to larger SP-frame spacings, while decoding error distortion decreased due to better bandwidth utilization. This agreed with our intuition. We also observed that larger loss rate induced larger decoding eror distortions, while drift distortions remained more or less the same. This observation is also intuitive.

For online optimization, a network simulator `muns`[5] is employed to simulate the behavior of four strategies: 1) `I16` - use of periodic I-frame every 16 frames with no auxiliary stream, 2) `rP` - redundant representation with regular P-frames in main stream and reconstructed P-frames in auxiliary stream, 3) `SP` - redundant representation with primary SP-frames in main stream, and secondary SP-frames in auxiliary stream, and 4) `rP+SP`, which is same as `SP`, except the auxiliary stream additionally contains reconstructed P-frames. Bandwidth was kept at $1.05\times$ the average bitrate of all I and P-frames, and the Gilbert channel at average burst length of 7 and loss rate of 5%. The results are given in Fig. 4(b). We see that while `I16` does not suffer from drift, it has mediocre average PSNR of 35.89 dB due to the high recovery time of 14.7 frames (not shown). Scheme `rP` has lowest recovery time of 3.39 frames, but suffer from drift, resulting in significant loss in PSNR compared to `rP+SP`. In contrast our proposed `rP+SP` scheme simultaneously achieve good average PSNR and short loss recovery time of about 3.5 frames. In particular, `rP+SP` out-performed `I16` in loss recovery time by 11.2 frames, and out-performed `rP` in PSNR by 2.2dB for $\Delta = 18$. In general, lower bit-cost associated with less frequent intra-frames would result in even larger recovery time for scheme based on intra-coding only.

7. CONCLUSION

In conclusion, we have proposed the use of reconstructed P-frames and secondary SP-frames as auxiliary stream and demonstrated it can achieve low recovery time and high average PSNR.

8. REFERENCES

[1] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," in *IEEE Transactions on Circuits and Systems for Video Technology*, July 2003, vol. 13, no.7.

[2] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, July 2003, vol. 13, no.7, pp. 560–576.

[3] W.-T. Tan, B. Shen, A. Patti, and G. Cheung, "Temporal propagation analysis for small errors in a single-frame in H.264 video," in *IEEE International Conference on Image Processing*, San Diego, CA, November 2008.

[4] P. Frossard and O. Verscheure, "Joint source/FEC rate selection for quality-optimal MPEG-2 video delivery," in *IEEE Transactions on Image Processing*, December 2001, vol. 10, no.12, pp. 1815–1825.

[5] G. Cheung, P. Sharma, and S.J. Lee, "Smart media striping over multiple burst-loss channels," in *IEEE Transactions on Selected Topics in Signal Processing*, August 2007, vol. 1, no.2, pp. 319–333.

[6] G. Cheung, N.-M. Cheung, and A. Ortega, "Optimized frame structure using distributed source coding for interactive multiview streaming," in *IEEE International Conference on Image Processing*, Cairo, Egypt, November 2009.

[7] G. Lawler, *Introduction to Stochastic Processes*, Chapman & Hall, 1995.