An Analytic Comparison of RPS Video Repair

Yubing Wang[*a], Mark Claypool[b], and Robert Kinicki[b]
[a]EMC Corporation, 32 Coslin Drive, Southborough, MA 01729, USA
[b]Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609, USA

ABSTRACT

Transmitting high-quality, real-time interactive video over lossy networks is challenging because network data loss can severely degrade video quality. A promising feedback technique for low-latency video repair is Reference Picture Selection (RPS), whereby the encoder selects one of several previous frames as a reference frame for predictive encoding of subsequent frames. RPS can operate in two different modes: an optimistic policy that uses negative acknowledgements (NACKs) and a more conservative policy that relies upon positive acknowledgements (ACKs). The choice between RPS NACK and ACK depends on the network conditions, such as round-trip time and loss probability, and the video content, such as low or high motion. This paper derives a set of analytical models to predict the quality of videos with RPS NACK or ACK. These models are used to study RPS performance under varied network conditions and with different video contents through a series of experiments. Analysis shows that the best choice of ACK or NACK greatly depends upon the round-trip time and packet loss, and somewhat depends upon the video content and Group of Pictures (GOP) size. In particular: 1) RPS ACK performs better than RPS NACK when round-trip times are low; 2) RPS NACK performs better than RPS ACK when the loss rate is low, and RPS ACK performs better than RPS NACK when the loss rate is high; 3) for a given round-trip time, the loss rate where RPS NACK performs worse than RPS ACK is higher for low motion videos than it is for high motion videos; 4) videos with RPS NACK always perform no worse than videos without repair for all GOP sizes; however, 5) below certain GOP sizes, videos without RPS outperform videos with RPS ACK. These insights derived from our models can help determine appropriate choices for RPS NACK and ACK under various network conditions and video contents.

Keywords: RPS, Reference Distance, PSNR, H.264

1. INTRODUCTION

Despite the improvements to networks and video display technologies, many network connections still lose data packets. Lost packets are especially detrimental to streaming video because of the dependency between video frames during encoding where one lost video packet can propagate errors to many subsequent video frames. While video clients can use local concealment to visually cover up transmission loss, the ability to adequately repair video without cooperation with the video server is limited. For video connections that have low end-to-end delay requirements between end hosts, such as video conferencing or interactive, thin-client desktops [1], feedback to the server to request retransmission of lost packets adds too much latency to the video session. Forward error correction [2] (FEC) has also been used to ameliorate the effects of loss, but FEC requires additional data to be added to the video stream and FEC encoding and decoding can be complicated.

A promising feedback technique for low-latency video repair is Reference Picture Selection (RPS) [3- 5], whereby the encoder uses one of several previous frames as a reference frame for encoding. Broadly, the reference frame can be the previous frame (called RPS NACK mode), as is done in typical video encoding, or the reference frame can be several frames older if the encoder waits for the receiver to confirm receipt of the frame (called RPS ACK mode). The difference in performance between RPS ACK and NACK mode depends upon the latency and loss pattern between the video sender and receiver and also upon the effects of reference distance on the encoded video quality. In particular, videos that do not significantly degrade with high reference distance benefit from the use of RPS ACK while videos that are sensitive to reference distance may choose RPS NACK or even some other repair techniques.

Although numerous studies have detailed the benefits of various repair schemes to video quality [5-7, 11, 12], to the best of our knowledge, there has been no systematic exploration of the impact of video and network conditions on the performance of RPS ACK and NACK. This paper derives two analytic models to predict the video quality of videos with RPS NACK or ACK. These models are then used to analyze RPS performance under various network conditions and video contents through a series of experiments. A set of high-quality videos with a wide variety of scene complexity

---

[*] wang_yubing@emc.com, phone 508-305-8849; claypool@cs.wpi.edu, phone 508-831-5409; rek@cs.wpi.edu, phone 508-831-6116

and motion characteristics are selected. These videos are all encoded using H.264 [8, 9], an increasingly popularly deployed compression standard with support for RPS, with a bandwidth constraint and a range of reference distances. Analysis shows that RPS performance is affected by a variety of factors including round-trip time, loss rate, video content and GOP size. In particular, analysis shows that:

- As round-trip times increase, the video quality for both RPS ACK and NACK degrades; however, RPS ACK performs better than RPS NACK when round-trip times are low.
- As loss rates increase, the video quality for both RPS ACK and NACK degrades; however, RPS NACK performs better than RPS ACK when the loss rate is low, and RPS ACK performs better than RPS NACK when the loss rate is high.
- For a given round-trip time, the loss rate where RPS NACK performs worse than RPS ACK is higher for low motion videos than it is for high motion videos.
- Videos with RPS NACK always perform no worse than videos without RPS for all GOP sizes; however, below certain GOP sizes, videos without RPS outperform videos with RPS ACK.

These insights derived from our models are useful for helping select the repair technique that maximizes video quality, particularly when choosing between RPS ACK and NACK modes.

The rest of this paper is organized as follows: Section 2 provides background information on technologies used in this paper; Section 3 provides the hypothesis for this study; Section 4 describes the analytical models for RPS in detail; Section 5 analyzes the experimental results; and Section 6 draws conclusions and describes possible future research.

2. BACKGROUND

This section provides background information on video technologies employed in this paper. Section 2.1 reviews H.264; and Section 2.2 discusses reference picture selection (RPS).

2.1 H.264

H.264, the state of the art in video compression standards [8, 9], is used throughout this study to encode/decode the video clips. The H.264 video codec supports a broad range of applications from low bit-rate Internet streaming applications to HDTV broadcast. An H.264 picture consists of macro-blocks, each containing 16x16 luminance samples, and two corresponding 8x8 chrominance samples. Within each picture, macro-blocks are arranged in slices, where a slice is a set of macro-blocks in raster scan order. The number of macro-blocks per slice need not be constant within a picture. In this paper, a slice containing a fixed number of successive macro-blocks is referred to as a GOB (Group of Blocks). Macro-blocks are classified as one of three types: I (intra-coded), P (predictive-coded) and B (bidirectional predictive-coded). I-blocks are encoded independently and contain all information required to decode the macro-block. P-blocks are encoded using motion compensation techniques which depend on the previous I or P macro-blocks. B-blocks further exploit motion compensation techniques by using information in the previous and following I or P macro-blocks. The H.264 encoder selects reference pictures for motion-compensated prediction [10] from several pictures that have been successfully decoded. However, as the reference distance increases, coding efficiency may degrade as similarities between the encoding frame and the reference frame decrease [13]. A P-block can be further divided into macro-block partitions; i.e. blocks of size 16x16, 16x8, 8x16 or 8x8 luminance blocks. By using these finer partitions for motion-compensated prediction, better prediction accuracy can be achieved.

2.2 Reference Picture Selection (RPS)

Transmitting high-quality, real-time interactive video over lossy networks is challenging because data loss can severely degrade video quality. Feedback-based error control techniques use information sent by the decoder to adjust the coding parameters at the encoder to achieve better error resilience [5]. With Reference Picture Selection (RPS) [3-5] the encoder selects one of several previous frames that have been successfully decoded as a reference for predictive encoding of subsequent frames. RPS operates in one of two modes. In negative acknowledgement (NACK) mode, in response to a transmission error, the decoder sends the encoder a NACK message for the erroneous frame and the number of a previously-received, correctly-decoded reference frame that can be used as a reference for prediction. Upon receiving the NACK, the encoder uses the indicated frame as a reference to encode the current frame. Figure 1 illustrates an example of RPS NACK mode. When the decoder observes that frame 4 has a transmission error, it sends a NACK to the encoder with an explicit request to use frame 3, which has been decoded correctly, for prediction. The NACK propagates back to the encoder, arriving before frame 7 is encoded. After receiving the NACK, the encoder then uses frame 3 as a reference to encode frame 7.

In RPS positive acknowledgement (ACK) mode, all correctly received frames are acknowledged and the encoder only uses acknowledged frames for reference. Since the encoder has to use an older (compared with NACK mode) reference frame for prediction, the coding efficiency of RPS ACK degrades as the round-trip time increases. However, RPS ACK mode entirely eliminates error propagation since the encoder only uses acknowledged frames as references. Figure 2 illustrates the use of RPS ACK mode. In this example, no transmission error occurs at first, allowing the encoder to receive an ACK for frame 1 while encoding frame 4, and thus uses frame 1 as a prediction reference to encode frame4. Similarly, the encoder uses frame 2 as a reference for frame 5, and frame3 as a reference for frame 6, and so on. However, since no ACK is received for frame 4, frame 7 uses acknowledged frame 3, instead of frame 4, as a reference frame.

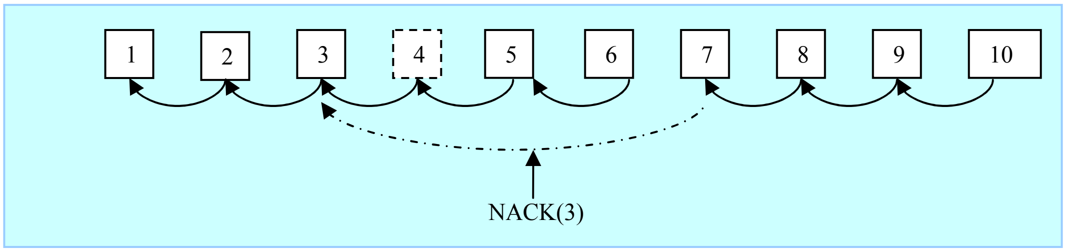


Fig. 1. Illustration of Encoding using RPS with NACK Mode. Frame 4 has a transmission error. The arrows indicate the selected reference frame for encoding.

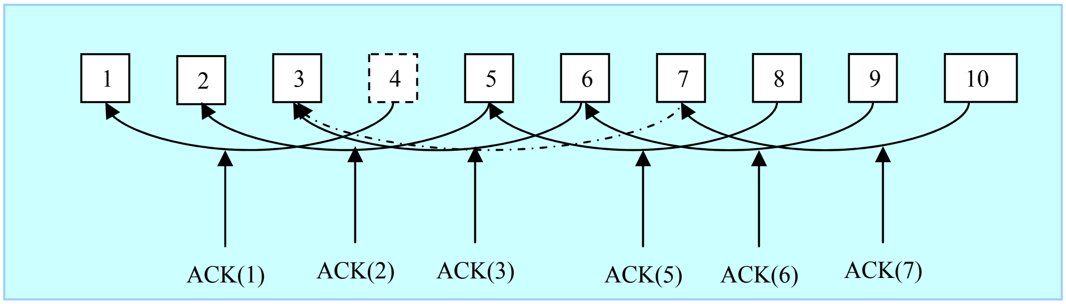


Fig. 2. Illustration of Encoding using RPS with ACK Mode. Frame 4 has a transmission error. The arrows indicate the selected reference frame.

3 HYPOTHESES

RPS selects one of several previous frames that have been successfully decoded as reference frame for prediction. The longer the round-trip time, the older the frame selected as a reference, and thus the lower video quality. RPS ACK always uses older frames as references even if no transmission errors occur, whereas RPS NACK uses the previous frame as a reference during error-free transmission. This informs our first two hypotheses:

Hypothesis 1: *RPS ACK performs better than RPS NACK when round-trip times are low.*

Hypothesis 2: RPS *NACK performs better than RPS ACK when loss rates are low.*

RPS NACK mode can maintain better coding performance during error-free transmission. However, if a transmission error occurs, the error propagates for a period of one round-trip time. The increase in loss rate increases the frequency of error propagation. This informs the third hypothesis:

Hypothesis 3: *RPS NACK performs worse than RPS ACK when loss rates are high.*

The cross-over point for loss rate below which RPS NACK outperforms RPS ACK decreases as round-trip time increases since the increase of error propagation does more harm to then video quality than does the increase of reference distance. Our previous study [13] shows low motion videos are more sensitive to a change in reference distance than are high motion videos. Thus, this cross-over point is affected by video content. This motivates our final hypothesis:

Hypothesis 4: *For a given round-trip time, the loss rate where RPS NACK performs worse than RPS ACK is higher for low motion videos than it is for high motion videos.*

4 ANALYTICAL MODEL

This section derives analytical models for RPS ACK and NACK, which aim at capturing the relationship between video quality with ACK or NACK under a variety of network characteristics including packet-loss rate, round-trip time and capacity constraints. These models are then used to test our hypotheses.

The models target H.264 videos since this standard incorporates RPS ACK and NACK, but can generally represent any video standard that uses RPS repair. The models assume the independent segment decoding (ISD) mode of H.264 where each GOB is encoded as an individual sub-video independently from other GOBs in the same frame, and the reference picture is selected on a per-GOB basis, i.e., for all macro-blocks within one GOB the same reference picture is used. Since errors inside a GOB do not propagate to other GOBs, the video sequence can be partitioned into independent video sub-sequences. We refer to an independent video sub-sequence as a *reference chain*, illustrated in Figure 3. Our model assumes that each GOB is carried in a single network packet.

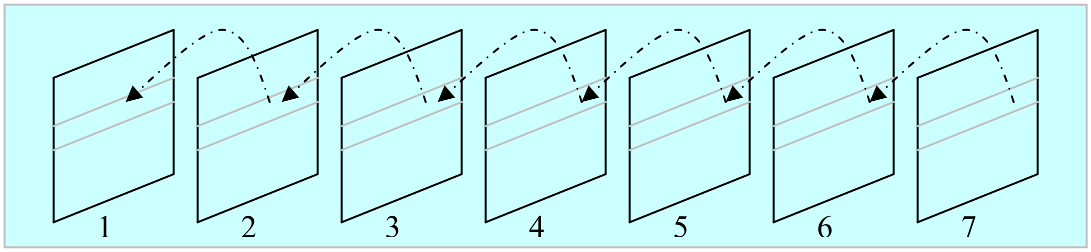


Fig. 3 Illustration of a reference chain, where each rectangle represents a video frame

Our models assume reliable transmission of feedback messages since feedback is usually not part of the video syntax and is transmitted via a separate network connection where control information is exchanged [5]. Such a connection may not suffer from congestion as does the forward link, or may use retransmission or other methods to ensure reliable delivery. The models also assume erroneously-decoded GOBs are locally concealed and make no assumption on specific local concealment techniques. Instead, constant quality for all locally concealed GOBs is assumed.

4.1 Model Parameters

Table 1 provides all the parameters used for our analytical models.

Table 1. Model Parameters

R_F	Encoded frame rate (in frames per second, fps - typical full-motion video frame rates are 25-30 fps)
N_G	GOP size (in frames)
t_{INT}	Time-interval between two frames (in milliseconds)
U_r	Average PSNR value for a GOB that is encoded using GOB that is *r* GOBs backward in reference chain as a reference
U_0	Average PSNR value for an Intra-coded GOB
U'	Average PSNR value for a GOB that is repaired using local concealment
t_{RTT}	Round-trip time (in milliseconds)
P	Packet loss probability (fraction)
C	Capacity constraint (in Mbps)
q_n	Probability that the *n*-th GOB in reference chain is decoded correctly
$q_{n,r}$	Probability of event that the *n*-th GOB in the reference chain is decoded successfully using the *r*-th GOB as a reference

Q_n	Expected PSNR value for *n*-th GOB in the reference chain

4.2 Analytical Model for RPS ACK

RPS ACK uses acknowledged frames as references. Since it takes at least a round-trip time for the encoder to receive an ACK for a GOB, the current GOB has to use a GOB which is at least δ[†] GOBs before it as a reference. The longer the round-trip time, the older the GOB that is selected as a reference, and thus the lower video quality.

Note that as long as GOB *n* is successfully received, it can be decoded successfully since it can use any previously-acknowledged GOB as a reference. Therefore, the probability of GOB *n* being successfully decoded is:

$$q_n = 1 - p \qquad (4.1)$$

Since the encoder selects the last GOB available without errors at the decoder as a reference, the reference GOB for GOB *n* could be chosen from GOB 1 up to GOB (*n*-*δ*). The probability of decoding GOB *n* correctly using GOB (*n*-δ-*i*) as a reference is:

$$(1-p)p^i q_{n-\delta-i}, \qquad 0 \le i \le n-\delta-1, \delta = \left\lceil \frac{t_{RTT}}{t_{INT}} \right\rceil \qquad (4.2)$$

Where $q_{n-\delta-i}$ is the probability of GOB (*n*-*δ*-*i*) being successfully decoded, p^i is the probability of *i* consecutive GOBs (preceding the GOB (*n*-*δ*)) having transmission errors and (1-*p*) is the probability of GOB *n* being successfully received.

The use of older reference GOBs for prediction degrades the effectiveness of compression for a GOB. Thus to maintain a constant frame rate and bit rate, the encoder uses a coarser quantizer and the overall video quality may decrease. To account for video quality degradation when using an older reference GOBs for prediction, U_r denotes the average PSNR for a GOB *n* whose reference GOB is *r* GOBs back in the reference chain.

The expected PSNR for the *n*-th GOB is as follows:

$$Q_n = \begin{cases} (1-p)\sum_{i=0}^{n-\delta-1} U_{\delta+i} p^i q_{n-\delta-i} + p * U', & n > \delta \\ (1-p)U_0 + p * U', & n \le \delta \end{cases} \qquad (4.3)$$

where U' denotes the average PSNR for a locally concealed GOB and U_0 the average PSNR for an intra-coded GOB. The values for U_i are obtained from our previous work [13]. Note that the first *δ* GOBs have to be encoded in intra mode since they cannot receive any ACK messages from the decoder.

Since $q_{n-\delta-i}$ is a constant (1-*p*), equation (4.3) can be further simplified as follows:

$$Q_n = \begin{cases} (1-p)^2 \sum_{i=0}^{n-\delta-1} U_{\delta+i} p^i + p * U', & n > \delta \\ (1-p)U_0 + p * U', & n \le \delta \end{cases} \qquad (4.4)$$

The average PSNR over an entire GOP can be computed as follows:

[†] $\delta = \left\lceil \frac{t_{RTT}}{t_{INT}} \right\rceil$

$$
\begin{aligned}
Q &= \frac{1}{N_G}\left(\sum_{n=1}^{\delta}((1-p)U_0 + pU^{'}) + \sum_{n=\delta+1}^{N_G}((1-p)^2\sum_{i=0}^{n-\delta-1}U_{\delta+i}p^i + pU^{'})\right) \\
&= \frac{1}{N_G}\left(((1-p)U_0 + pU^{'})\delta + \sum_{n=\delta+1}^{N_G}((1-p)^2\sum_{i=0}^{n-\delta-1}U_{\delta+i}p^i + pU^{'})\right)
\end{aligned}
\qquad (4.5)
$$

4.3 Analytical Model for RPS NACK

For RPS NACK mode, one of the GOBs in the previous frame is used as a reference GOB during the error-free transmission. After a transmission error, the decoder sends a NACK for the erroneous GOB with an explicit request to use older, intact GOBs as a reference. Therefore, the encoder may use a GOB in the previous frame or one in older frames as a reference to encode the current GOB *n* depending upon whether it receives a NACK from the decoder or not. If a NACK is not received from the decoder, the encoder uses a GOB in the previous frame as a reference. The probability of correctly decoding GOB *n* using a GOB in the previous frame as reference is denoted as $q_{n,1}$, where 1 indicates using the preceding GOB in the reference chain as a reference. If the encoder does receive a NACK, it uses the GOB requested by the decoder as a reference. As in ACK mode, the reference GOB for GOB *n* could be chosen from GOB 1 up to GOB (*n*-δ) depending upon which GOB is the last correctly decoded GOB. $q_{n,\delta+i}$ $(0 \le i \le n-\delta-1)$ denotes the probability of decoding GOB *n* correctly using GOB (*n*-δ-*i*) as a reference. Since none of the first δ GOBs receives a NACK before being encoded, the successful decoding of each subsequent GOB depends upon the success of the preceding GOBs. Therefore, the probability of GOB *n* being successfully decoded is as follows:

$$
q_n = \begin{cases} q_{n,1} + \sum_{i=0}^{n-\delta-1} q_{n,\delta+i}, & n > \delta \\ (1-p)^n, & n \le \delta \end{cases}
\qquad (4.6)
$$

The expected PSNR for GOB *n*:

$$
Q_n = \begin{cases} U_1 q_{n,1} + \sum_{i=0}^{n-\delta-1} U_{\delta+i} q_{n,\delta+i} + (1-q_n)U^{'}, & n > \delta \\ (1-p)^n U_1 + (1-(1-p)^n)U^{'}, & 1 < n \le \delta \\ \bigl(1-p)*U_0 + p*\bigr)U^{'}, & n = 1 \end{cases}
\qquad (4.7)
$$

The average PSNR over a GOP can be computed as follows:

$$
Q = \frac{1}{N_G}\left(\sum_{n=1}^{\delta}((1-p)^n U_1 + (1-(1-p)^n)U^{'}) + \sum_{n=\delta+1}^{N_G}(U_1 q_{n,1} + \sum_{i=0}^{n-\delta-1} U_{\delta+i} q_{n,\delta+i} + (1-q_n)U^{'})\right)
\qquad (4.8)
$$

4.3.1 GOB Dependency Modeling

To estimate $q_{n,1}$ and $q_{n,\delta+i}$ ($0 \le i \le n-\delta-1$), it is essential to model the prediction dependency between GOBs in the reference chain. A binary tree is adopted to model GOB dependency for RPS with NACK mode. Two input parameters are required to build the dependency tree: packet loss probability (*p*) and round-trip time (δ). Figure 4 illustrates a binary tree for the possible decoded versions of a GOB and the corresponding reference GOB selections while using RPS with NACK mode. In the illustrated example, there are four GOBs and the round-trip time equals two GOBs.

A node in the tree represents a decoded version of a GOB in a video frame. The nodes with hollow circles are those decoded erroneously while those with solid circle are those decoded correctly. Branches leaving a node represent the two cases that either a packet[‡] is received erroneously with probability *p* or received correctly with probability (1-*p*). For a node whose decoded status is "Erroneous", the number (in parenthesis) besides the node represents the last GOB that has been decoded correctly at the decoder when entering this node; for a node with "Correct" decoded status, the number (in brackets) represents its reference GOB number. The GOB number between two dashed lines represents the GOB in transmission. Note that the root node (labeled with a crossed circle) represents an intra-coded GOB. Therefore, each time the encoder intra-codes a GOB, the binary tree will be refreshed.

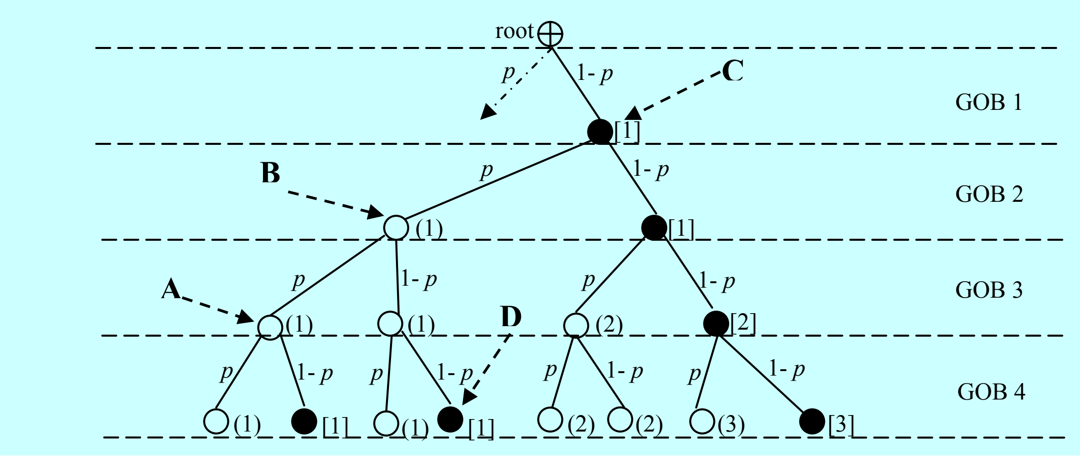


Fig. 4 Binary Tree for the possible decoded versions of a GOB and the corresponding reference frame selections using RPS with NACK mode[§].

To illustrate how the decoded status of a node is decided, consider node A (GOB 3). Since the ancestor of node A (node C) is received correctly, node A (GOB 3) did not receive a NACK message from the decoder and thus used its parent (node B, GOB 2) as a reference. Since the decoded status of node B is "Erroneous", the decoded status of node A is "Erroneous" as well. Upon entering node A, the last correctly decoded GOB at the decoder is GOB 1, therefore, the number (in parenthesis) besides node A is 1. Next, consider node D (GOB 4). Since D's ancestor 2 frames back (node B, GOB 2) is received erroneously (under the *p* branch), node B receives a NACK message from the decoder, which explicitly requires using GOB 1 as a reference. Thus, the reference GOB for node B is GOB 1. Furthermore, node B is under the (1-*p*) branch to indicate GOB 4 was correctly received. Therefore, the decoded status of node B is "Correct".

4.3.2 GOB Dependency Tree Creation

Each node in the GOB dependency tree contains the following information:

- *Decoded Status*: Correct or Erroneous.
- Probability of occurrence of this node (decoded version of a GOB).
- *Latest GOB that has been decoded correctly at the decoder when entering this node* (LDC[**] for short) - this information is recorded only for a node with "Erroneous" decoded status.
- *Reference GOB* - this information is recorded only for a node with "Correct" decoded status.
- GOB number in the reference chain.

The creation of the GOB dependency tree begins with a successfully decoded intra GOB. Since no correctly decoded GOBs are under the *p* branch of the root node, that part of the tree can be ignored. For each node, the four parameters described above are determined using the following algorithm:

1. If this node is under the *p* branch:

[‡] As stated earlier, our model assumes each packet contains one GOB.

[§] The first *p* branch and its descendants are not shown because there are no correctly decoded GOBs under this branch.

[**] LDC stands for Latest Decoded Correctly

- Its decoded status is set to “Erroneous”;
- Its probability is set to: $p * parent \rightarrow prob$, where $parent \rightarrow prob$ is its parent’s probability;
- Its LDC is determined based upon its parent’s decoded status. If its parent is decoded correctly, its LDC is set to its parent’s GOB number; otherwise the LDC is set to its parent’s LDC;
- Its GOB number is set to its parent’s GOB number plus 1.

2. If this node is under the (1- p) branch:
    - Its decoded status is determined based upon whether its ancestor δ frames back is received correctly and the decoded status of its parent. If its ancestor δ frames back was received correctly (no NACK), then check its parent’s decoded status. If its parent is decoded correctly, its decoded status is set to “Correct”, otherwise it is set to “Erroneous”. If its ancestor δ frames back was *not* received correctly, its decoded status is set to “Correct” since it received a NACK message from the decoder and used an older, correctly decoded GOB as a reference.
    - Its probability is set to: $(1-p) * parent \rightarrow prob$, where $parent \rightarrow prob$ is its parent’s probability;
    - If its decoded status is “Correct”, its reference GOB is determined based upon the decoded status of its ancestor δ frames back. If its ancestor δ frames back was received correctly (no NACK), its reference GOB is set to its parent’s GOB number. If its ancestor δ frames back was NOT received correctly, its reference GOB is set to its ancestor’s LDC.
    - If its decoded status is “Erroneous”, its LDC is determined based upon the decoded status of its parent. If its parent was decoded correctly, its LDC is set to its parent’s GOB number; otherwise it is set to its parent’s LDC.
    - Its GOB number is set to its parent’s GOB number plus 1.

4.3.3 Estimate of $q_{n,r}$ using the GOB Dependency Tree

After building the GOB dependency tree, $q_{n,r}$ is estimated in two steps. First, the GOB dependency tree is traversed to find all “Correct” nodes with GOB number equal *n* and reference GOB number (*n-r*). Then the probabilities of each node from step 1 are added together to produce an estimate for $q_{n,r}$.

5 ANALYSIS

This section analyzes RPS performance under various network conditions and video contents. The results presented in this section are obtained through a series of experiments using the analytical models described in Section 4. These experiments select a set of video clips with a variety of motion content. Each video sequence contains 300 video frames with a frame rate of 25 frames/second (fps). These videos are all encoded using H.264 with a bandwidth constraint and a range of reference distances. The content of these video clips can be roughly categorized into three groups: high motion/scene complexity, medium motion/scene complexity and low motion/scene complexity. Table 2 provides an approximate content classification of each video clip, with an identifying name and a short description of the video content.

Table 2. Video Clips Used in the Experiments.

Video Clip	Motion	Description
Container	Low	A container ship moving slowly
News	Low	Two news reporters talking
Silent	Medium	A person demonstrating sign language
Mom & Daughter	Medium	A mother and daughter talking
Foreman	High	A foreman talking
Mobile	High	Panning of toy train moving

We first examine the impact of round-trip time on RPS video quality. Figure 5 depicts PSNR versus round-trip time for videos with RPS NACK under varying loss rates. The video clip for this experiment is *news*[††] the GOP size is 22. As round-trip time increases, in Figure 5, average PSNR degrades for all loss rates. However, the degrees of quality degradations are not uniform. Clearly with RPS NACK, video quality under higher packet loss degrades faster than under lower packet loss. For RPS NACK, each transmission error propagates for a period of one round-trip time. Thus, increased packet loss induces more frequent GOB error propagation and video quality degrades quickly.

Figure 6 depicts PSNR versus round-trip time for videos with RPS ACK under varying loss rates. As the round-trip time increases, similar to RPS NACK, the average PSNR for videos with RPS ACK degrades for all loss rates. However, unlike with RPS NACK, RPS ACK video quality degrades slower under higher packet loss than under lower packet loss. When the packet loss rate is low, the major cause of video quality degradation for RPS ACK is increased reference distance (i.e., the round-trip time); whereas under higher packet loss rates, the video quality degradation for RPS ACK is attributed more to packet loss than to round-trip time.

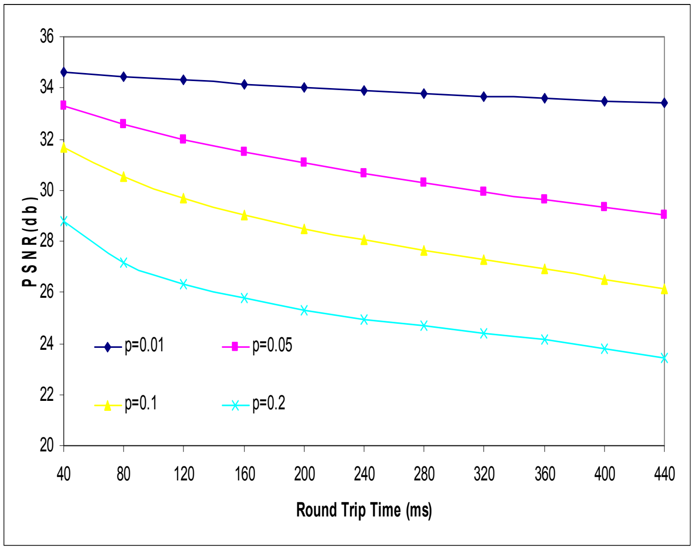


Fig. 5 PSNR vs. RTT with RPS NACK under Different Loss

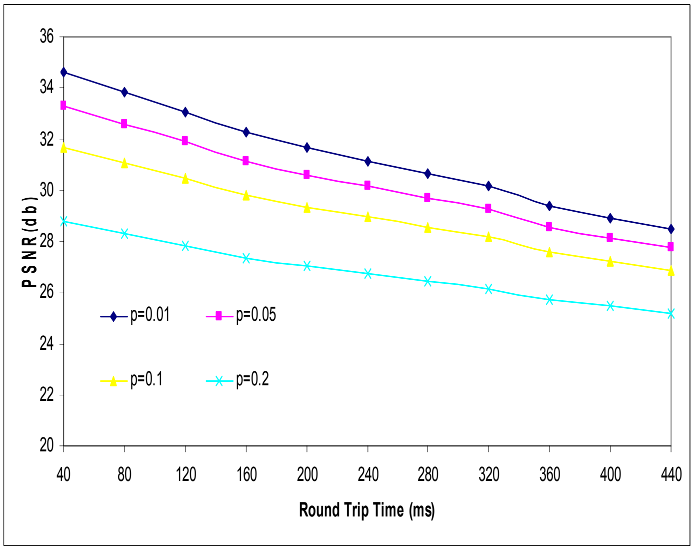


Fig. 6 PSNR vs. RTT with RPS ACK under Different Loss

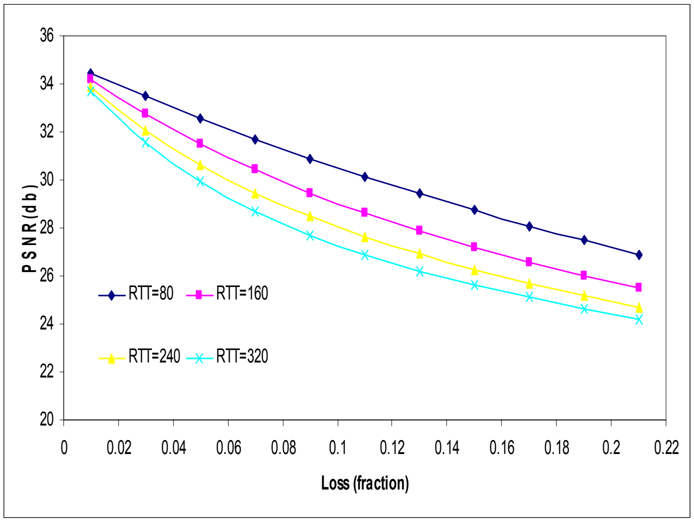


Fig. 7 PSNR vs. RTT with RPS NACK under Different Loss

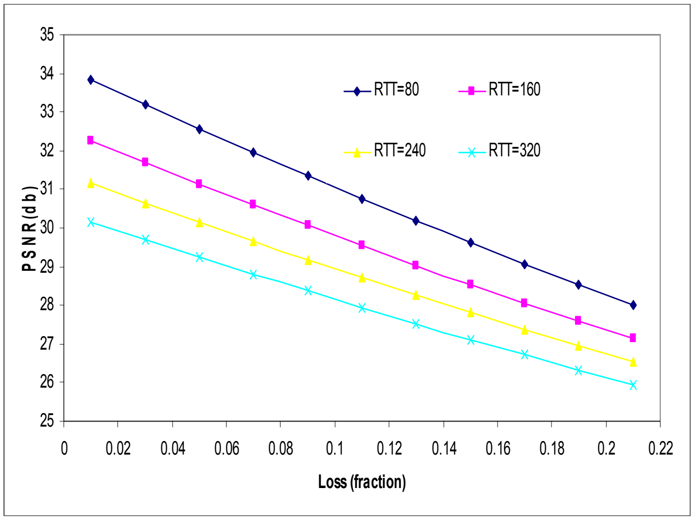


Fig. 8 PSNR vs. RTT with RPS ACK under Different Loss

[††] In this clip, two news reporters are talking with somebody dancing in the background

Next we investigate how RPS video quality changes as loss probability is varied. Figure 7 provides PSNR versus loss probability curves for a video encoded with RPS NACK for four round-trip times. As the loss probability increases, average PSNR using RPS NACK degrades for all round-trip times. However, with RPS NACK, video quality under higher round-trip times degrades faster than under lower round-trip times. This is due to the fact that with RPS NACK, larger round-trip time implies longer error propagation periods that causes video quality to degrade. Figure 8 graphs PSNR versus loss probability curves for a video encoded with RPS ACK for the same four round-trip times. As loss probability increases, like RPS NACK, for RPS ACK the average PSNR degrades for all round-trip times. However, unlike RPS NACK, for RPS ACK, the video quality under higher round-trip times degrades slower than those under lower round-trip times. Under higher RTTs, video quality degradation for RPS ACK is attributed more to the round-trip time than to the packet loss, whereas under lower round-trip times, packet loss is the dominant cause of video quality degradation.

Similar trends were observed for six other tested video clips representing a variety of motion characteristics. Due to space limitations, those results are not presented in this paper.

The prior analysis demonstrates that RPS video quality for both ACK and NACK are affected by round-trip time and packet loss. To make informed repair technique choices, it is useful to know the range of packet loss within which RPS NACK performs better than RPS ACK or vise versa, and how the cross-over point changes with round-trip time and video content. Figures 8-10 compare RPS NACK and ACK under different round-trip times by graphing PSNR versus packet loss. All three experiments use the *news* video clip. As shown in Figure 8, with an 80 ms round-trip time, when loss probability is less than 0.092, RPS NACK outperforms RPS ACK and when loss probability is larger than 0.092, RPS NACK performs better than RPS ACK. When round-trip time is increased to 160 ms in Figure 9, the cross-over point is reduced to 0.082. Furthermore, in Figure 10 with round-trip time 500 ms, the cross-over point is further reduced to 0.075. This suggests that as round-trip time increases, the video quality with RPS NACK degrades faster than RPS ACK. For RPS NACK, increased round-trip time produces longer GOB error propagation; whereas for RPS ACK, increased round-trip time yields higher GOB reference distances. Note, increasing error propagation does more harm to video quality than increasing reference distance.

We further investigate how the relationship between cross-over point and round-trip time is affected by video content. Figure 11 shows cross–over point versus round-trip time for six distinct videos. For loss rates above the trend-lines, RPS ACK performs better than RPS NACK. For loss rates below the trend-lines, RPS NACK performs better than RPS ACK. As round-trip time is increased, all the video cross-over points are lowered. This suggests that regardless of video content, increasing the error propagation is more harmful to video quality than increasing reference distance. For a fixed round-trip time, the cross-over points for low-motion videos are higher than those for high-motion videos. This implies that RPS ACK outperform RPS NACK over a wider range of packet loss for high-motion videos than for low-motion videos. This is primarily due to the fact that high-motion videos are less sensitive to the change of reference distance and thus can achieve better video quality with RPS ACK.

Finally, the RPS ACK and NACK models are used to investigate the impact of GOP size on video quality. Figure 12 graphs average PSNR versus GOP size for videos with RPS NACK for four round-trip times and the video with no repair. The loss probability for this experiment is 0.05. Below GOP size of 5, PSNR increases in all cases. After GOP size reaches 5, the PSNR for the video without RPS degrades due to error propagation. With RPS NACK, when round trip times are 80 ms and 160 ms, PSNR increases and becomes asymptotically steady. When round trip times are 240 ms and 320 ms, PSNR first slightly decreases and becomes asymptotically steady. For all GOP sizes, videos with RPS NACK perform no worse than videos without RPS; and RPS NACK performs better under lower round-trip times than under higher round-trip times since higher round-trip times introduces longer periods of error propagation.

Figure 13 depicts average PSNR versus GOP size for videos with RPS NACK under various round-trip times. As GOP size increases, PSNR increases for videos with RPS ACK for all round-trip times shown. Since RPS ACK uses intra coding before any GOBs are acknowledged, the PSNR for the first part of the GOB chain remains constant and increases only after ACKs are received by the encoder.. For all GOP sizes, RPS ACK performs better under lower round-trip times than under higher round-trip times since RPS ACK under higher round-trip times has to use older GOBs as reference for prediction. Note that below a certain GOP size, video without RPS actually performs better than videos with RPS ACK. For instance, when round-trip time is 80 ms and GOP size is below 8, videos without RPS performs better than videos with RPS ACK. This is because videos without RPS always use the previous GOB as reference and rely on intra coding to stop error propagation. When the GOP size is small, error propagation can be stopped very quickly, whereas, RPS

ACK always uses older GOBs as reference. Therefore, when the loss probability is low and GOP size is small, videos without RPS outperform videos with RPS ACK.

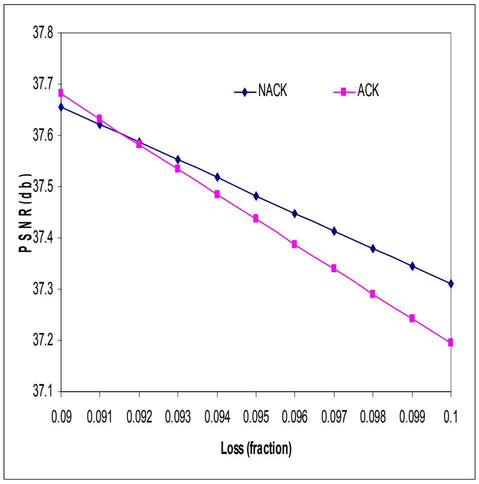


Fig. 8 RPS NACK vs. ACK with RTT= 80ms

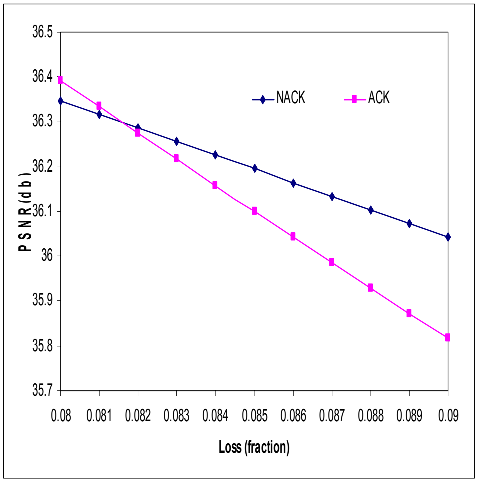


Fig. 9 RPS NACK vs. ACK with RTT= 160ms

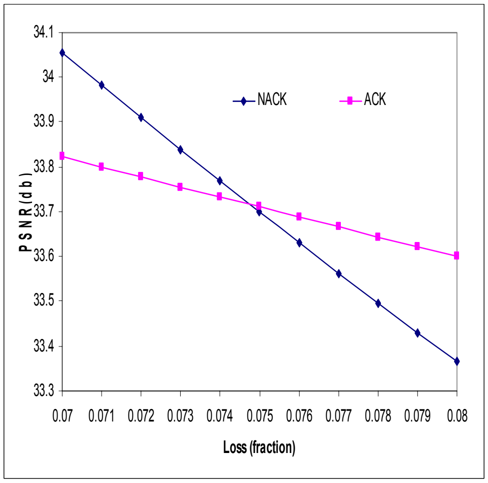


Fig. 10 RPS NACK vs. ACK with RTT= 400ms

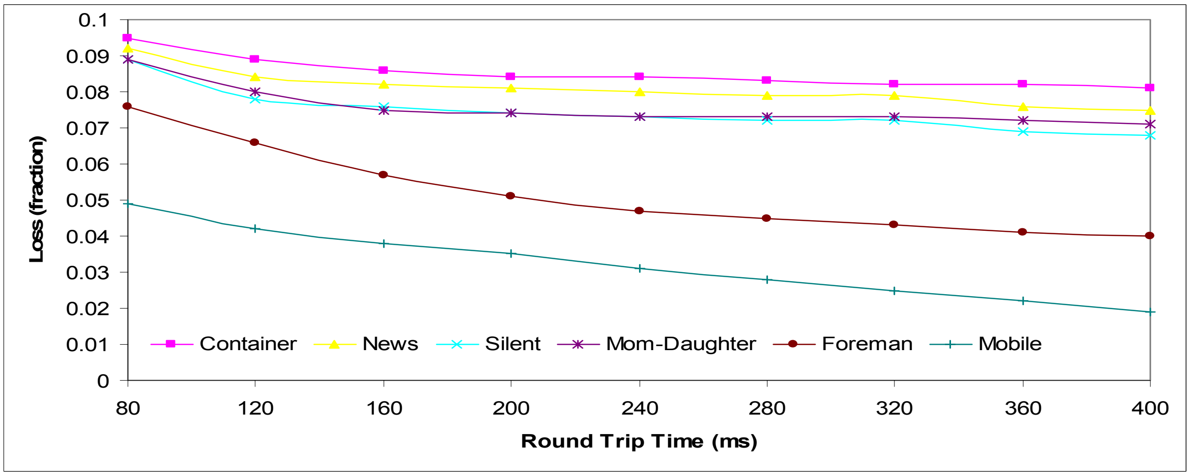
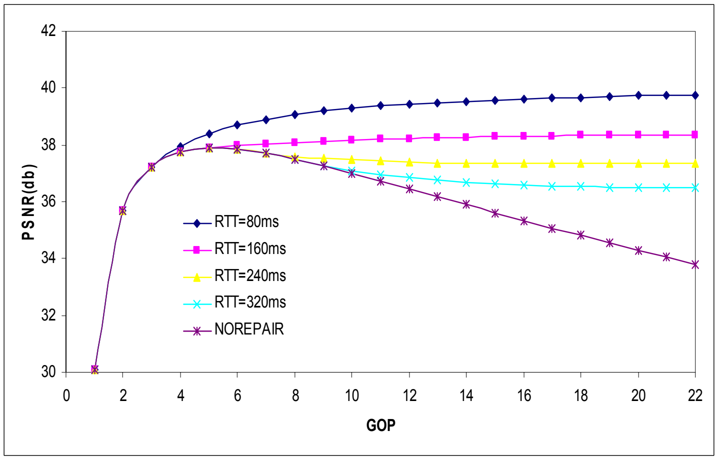


Fig. 11 The Cross-Over Point for Loss vs. Round-Trip Time for Six Video Clips



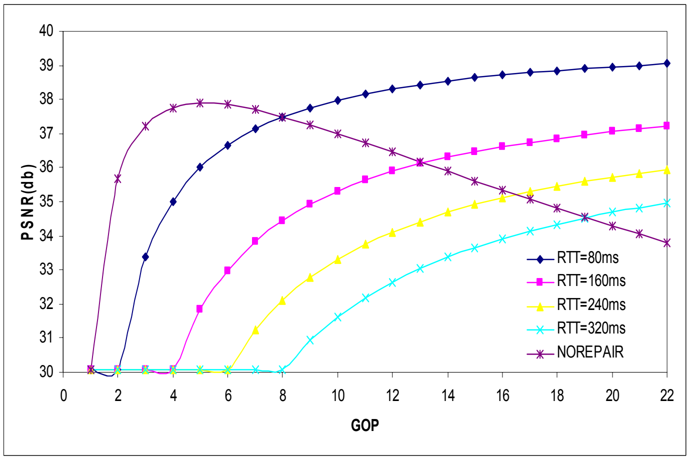


Fig. 12 PSNR vs. GOP with RPS NACK

Fig. 13 PSNR vs. GOP with RPS ACK

6 CONCLUSION AND FUTURE WORK

Reference Picture Selection (RPS) is an established video repair technique that allows the encoder to select one of several previous GOBs that have been successfully decoded as a reference GOB for predictive encoding of subsequent GOBs. RPS operates in either ACK or NACK mode. RPS NACK uses the previous GOB as reference during error-free transmission and uses an older GOB as reference when an error occurs. RPS NACK cannot eliminate error propagation since a packet loss results in an error propagation for about a round-trip time. RPS ACK always uses acknowledged GOBs as reference and thus eliminates error propagation entirely. However, using an older GOB as a reference reduces coding efficiency and results in lower video quality. Therefore, both RPS NACK and ACK have merits and drawbacks. The choice between RPS NACK and ACK depends on network conditions, such as round-trip time loss probability and video content. This paper compares RPS NACK and ACK under various network conditions and video content using two analytical models. Analysis using the models shows:

RPS ACK is more sensitive to round-trip times whereas RPS NACK is more sensitive to packet loss; for a given round-trip time, the loss rate where RPS NACK performs worse than RPS ACK is higher for low motion videos than it is for high motion videos; Videos with RPS NACK always perform no worse than videos without RPS for all GOP sizes. However, below certain GOP sizes, videos without RPS outperform videos with RPS ACK.

Our study uses PSNR to measure video quality. Future work could continue our current work with the reportedly more accurate Video Quality Metric (VQM) [14]. The analytical models presented in this paper and the results derived for them are yet to be verified through simulations and user studies.

REFERENCES

1. D. De Winter, P. Simoens, L. Deboosere, F. DeTurck, J. Moreau, B. Dhoedt and P. Demeester. "A Hybrid Thin-Client Protocol for Multimedia Streaming and Interactive Gaming Applications", *Proceedings of the Network and Operating System for Digital Audio and Video Workshop (NOSSDAV)*, Newport, RI, USA, May 2006.
2. J.C. Bolot, S. Fosse-Parisis, and D. Towsley. "Adaptive FEC-Based Error Control for Internet Telephony". *Proceedings of IEEE INFOCOM'99*, pages 1453–1460, Mar. 1999.
3. S. Fukunaga, T. Nakai, and H. Inoue. "Error Resilient Video Coding by Dynamic Replacing of Reference Pictures", *Proceedings of IEEE Global Telecommunications Conf. (GLOBECOM)*, London, vol. 3, Nov. 1996, pp.1503–1508.
4. Y. Tomita, T. Kimura, and T. Ichikawa. "Error Resilient Modified Inter-frame Coding System for Limited Reference Picture Memories", In *Proceedings of Int. Picture Coding Symp. (PCS)*, Berlin, Germany, Sept. 1997, pp.743–748.
5. B. Girod, and N. Färber. "Feedback-Based Error Control for Mobile Video Transmission". In *Proceedings of IEEE*, *Special Issue on Video for Mobile Multi-media*, vol. 97, no. 10, pp. 1707-1723, Oct. 1999.
6. Y. Wang, Q. Zhu. "Error Control and Concealment for Video Communication: A Review". In *Proceedings of the IEEE*, *Special Issue on Video for Mobile Multi-media,* Vol. 86, No. 5, May 1998.
7. Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos. "Error Resilient Video Coding Techniques," *IEEE Signal Proc. Mag.*, Vol. 17, pp. 61-82, Jul. 2000.
8. I. E. G. Richardson. "H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia", *Wiley*, ISBN 0-470-84837-5, 2004.
9. Joint Video Team of ITU-T and ISO/IEC JTC 1. "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)," document JVT-G050r1, May 2003.
10. T. Wiegand, X. Zhang, and B. Girod. "Long-Term Memory Motion-Compensated Prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, No. 1, pp. 70-84, Feb. 1999.
11. K. Mayer-Patel, L. Le, and G. Carle. "An MPEG Performance Model and Its Application to Adaptive Forward Error Correction", *Proceedings of ACM Multimedia*, Juan-les-Pins, France, Dec. 2002.
12. H. Wu, M. Claypool, and R. Kinicki. "A Model for MPEG with Forward Error Correction and TCP-Friendly Bandwidth", *Proceedings of Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Monterey, California, USA, June 2003.
13. Y. Wang, M. Claypool and R. Kinicki. "Impact of Reference Distance for Motion Compensation Prediction on Video Quality", *Proceedings ACM/SPIE Multimedia Computing and Networking (MMCN) Conference*, San Jose, California, USA, Jan. 2007.
14. M. Pinson and S. Wolf. "A New Standardized Method for Objectively Measuring Video Quality," *IEEE Transactions on Broadcasting*, Vol. 50, No. 3, pp. 312-322, Sep. 2004.