On Combining Temporal Scaling and Quality Scaling for Streaming MPEG

Huahui Wu, Mark Claypool, and Robert Kinicki
Computer Science Department
Worcester Polytechnic Institute
100 Institute Rd, Worcester, MA 01609, USA
{flashine|claypool|rek}@cs.wpi.edu

ABSTRACT

Temporal Scaling and Quality Scaling are both widely-used techniques to reduce the bitrate of streaming video. However, combinations and comparisons of Temporal and Quality Scaling have not been systematically studied. This research extends previous work to provide a model for combining Temporal and Quality Scaling, and uses an optimization algorithm to provide a systematic analysis of their combination over a range of network conditions and video content. Analytic experiments show: 1) Quality Scaling typically performs better than Temporal Scaling, with performance differences correlated with the motion characteristics of the video. In fact, when the network capacity is moderate and the loss rate is low, Quality Scaling performs nearly as well as the optimal combination of Quality and Temporal Scaling; 2) when the network capacity is low and the packet loss rate is high, Quality Scaling alone is ineffective, but a combination of Quality and Temporal Scaling can provide reasonable video quality; 3) adjusting the amount of Forward Error Correction (FEC) provides significantly better performance than video streaming without FEC or video streaming with a fixed amount of FEC.

1. INTRODUCTION

Temporal Scaling and Quality Scaling are commonly used to scale back real-time streaming video data rates to adjust to a capacity constraint, caused by the Internet Service Provider (ISP)'s negotiated rate, or to be TCP-Friendly [2]. With Quality Scaling, the video server adjusts the quantization level before transmission to reduce the streaming bitrate. With Temporal Scaling, the server discards some of the video frames before transmitting them over the network. While many researchers have studied Temporal Scaling or Quality Scaling [3, 6, 8, 14, 15] and many commercial video streaming products have incorporated these scaling methods, to the best of our knowledge there have been no systematic study comparing and combining Temporal Scaling and Quality Scaling.

NOSSDAV '06 Newport, Rhode Island USA

Besides adjusting to capacity constraints, video streaming must also respond to packet loss. While video streaming applications can typically tolerate reduced data through scaling, even small amounts of packet loss can produce unacceptably low video quality. Since video encoding relies upon inter-frame dependencies to achieve high compression rates, the random dropping of packets by routers or the random loss of frames on a wireless link can seriously degrade video quality. For example, as little as 3% MPEG packet loss can cause 30% of the frames to be undecodable.

To avoid the latency and variance in latency caused by the retransmission of lost packets, streaming video flows can use Forward Error Correction (FEC) to reconstruct lost streaming video packets. However, FEC-based repair techniques inherently add redundant data to the original video stream. When a streaming video operates within a capacity constraint, adding FEC reduces the effective transmission rate of the original video content [7]. Hence, selecting the right amount of FEC along with a complementary scaling approach (Temporal or Quality) can be cast as a constrained optimization problem that attempts to optimize the quality of the video stream.

Our previous research focused on the impact of FEC on video streaming, using first Temporal Scaling [15] and then Quality Scaling [14], to stream under a capacity constraint. Our results showed that FEC is critical for acceptable performance and works best when dynamically adjusted to the current network packet loss rate and capacity constraint. Either Temporal or Quality Scaling provides acceptable methods of scaling, but the research made no attempt at comparing, much less combining, the two scaling approaches. The focus of this paper is on the combination of both Temporal and Quality scaling. Specifically, our previously developed analytic model is extended to characterize the performance of MPEG[1] video with the combination of Temporal Scaling and Quality Scaling. This new model incorporates both a Temporal and Quality Scaling level and adjusts the number of FEC packets for each MPEG frame type. From the MPEG characteristics, video distortion is approximated using the scaling parameters and the video frame loss rate. Then, a new optimization algorithm is built to exhaustively search all possible combinations of scaling levels and FEC patterns to find the configuration that yields the best video quality under the capacity constraint.

Experimental results indicate that with moderate capacity constraints and loss rates, combining Temporal Scal-

[1]MPEG-1 is used throughout this paper, but the analysis and the results should be independent to the video standard.

ing with Quality Scaling provides very little improvement to video quality over using only Quality Scaling. Quality Scaling works better than Temporal Scaling under most circumstances and their performance difference is correlated to the video's motion characteristics. However, when available capacity is low and loss rate is high, Temporal Scaling and Quality Scaling used in combination yield significantly better video quality than Quality Scaling alone. The results also show that Adjustable FEC provides significant performance improvement over video streaming without FEC or video streaming with a fixed amount of FEC.

The remainder of the paper is organized as follows: Section 2 provides background knowledge for the work in this paper; Section 3 introduces the analytical model for computing video quality; Section 4 presents the optimization algorithm; Section 5 discusses experiments and analyzes the results; and Section 6 summarizes our conclusions.

2. BACKGROUND

2.1 Temporal Scaling

In *Temporal Scaling*, lower priority video frames are discarded prior to the Group of Picture (GOP) transmission. For instance, with the GOP pattern of '`IBBPBBPBBPBBPBB`', the data rate can be reduced by discarding all the B frames and only sending '`I--P--P--P--P--`'.

While Temporal Scaling could, in theory, select any of the frames in a GOP to discard, the MPEG frame dependencies limit the practical choice of frames to discard. Table 1 lists all the scaling levels used in this paper, accounting for the MPEG frame dependencies and minimizing the effect of Temporal Scaling on the quality of the received video. In the table, N_{PD} and N_{BD} are defined as the number of P or B frames transmitted in one GOP, respectively, with the scaling patterns provided for each scaling level, l_{TS}. Since typical MPEG decoders detect, and accommodate, lost frames, the frames selected for discarding can be removed at the sender without additional overhead.

Scaling Lvl (l_{TS})	N_{PD}	N_{BD}	Scaling Pattern
0	4	10	`IBBPBBPBBPBBPBB`
1	4	5	`IB-PB-PB-PB-PB-`
2	4	0	`I--P--P--P--P--`
3	0	0	`I--------------`

Table 1: Temporal Scaling Levels

2.2 Quality Scaling

To meet a capacity constraint, *Quality Scaling* uses higher quantization values to encode each MPEG frame with lower precision and fewer bits by removing low order bits from each DCT coefficient.

This paper assumes a SPEG (Scalable MPEG) [6] quality scaling model, with every DCT coefficient divided into four layers: one base layer and three advance layers. A DCT coefficient, C, is partitioned into the layers as follows:

$$\begin{array}{ll} \textit{Base Layer } L0: & C_0 = C >> 3 \\ \textit{1st Advance Layer } L1: & C_1 = (C >> 2)\&1 \\ \textit{2nd Advance Layer } L2: & C_2 = (C >> 1)\&1 \\ \textit{3rd Advance Layer } L3: & C_3 = C\&1 \end{array} \quad (1)$$

At the receiver/player side, the above steps are reversed to reconstruct the original MPEG video where zero is used instead when some advance layer(s) is (are) absent. This is analogous to using a high quantization value during MPEG encoding. Assuming the highest quantization value used is 3 (this yields a high fidelity quality and reasonable bitrate), it is not difficult to define the relationship of the scaling level, transmitting SPEG layers and equivalent quantization value as in Table 2. Since SPEG needs to use extra headers to indicate layer information (a 15%-25% overhead is indicated by the author of [6]), 20% overhead is used in this paper.

Scaling Lvl (l_{QS})	SPEG Layers	Equ. Quan. Val. v_Q
0	L0+L1+L2+L3	3
1	L0+L1+L2	6
2	L0+L1	12
3	L0	24

Table 2: Quality Scaling Levels

2.3 Forward Error Correction (FEC)

Forward Error Correction (FEC) can be used to recover video frames damaged by packet loss. Reed-Solomon (R-S) code [11] is a typical FEC technique that can be applied at the packet level. An application level video frame can be modeled as being transmitted in K packets, R-S adds $(N-K)$ redundant packets to the K original packets and sends the N packets. Although some packets may be lost, the frame can be completely reconstructed if any K or more packets are successfully received.

To analyze the effects of FEC on application layer frames, the sending of packets is modeled as a series of independent Bernoulli trials. Thus, the probability $q(N,K,p)$ that a K-packet video frame is successfully transmitted with $N-K$ redundant FEC packets along a network path with packet loss probability p is:

$$q(N,K,p) = \sum_{i=K}^{N} \left[\binom{N}{i} (1-p)^i * p^{N-i} \right] \quad (2)$$

Given I, P, and B frame sizes, and the distribution of redundant FEC packets added to each frame type, Equation 2 provides the probability of successful transmission for each frame type.

3. MODEL

3.1 Parameters and Variables

Layer	Parameters
MPEG	G, S_I, S_P, S_B
Scaling	$l_{TS}, N_{PD}, N_{BD}, l_{QS}, v_Q$
Repair(FEC)	S_{IF}, S_{PF}, S_{BF}
Network	p, t_{RTT}, s, T

Table 3: System Layers and Parameters

The system layers and parameters for our analytic model are indicated in Table 3, where the parameters are:

G: the GOP rate (in GOPs per second) during encoding

S_I, S_P, S_B: the size (in fixed-sized packets) of I, P or B frames, respectively, depending on quantization value v_Q.

l_{TS}: the Temporal Scaling level, as in Table 1.

N_{PD}, N_{BD}: the number of P or B frames, respectively, sent per GOP after Temporal Scaling, depending on l_{TS} as in Table 1.

l_{QS}: the Quality Scaling level, as in Table 2.

v_Q: the quantization value, depending on l_{QS} as in Table 2.

S_{IF}, S_{PF}, S_{BF}: the number of FEC packets added to each I, P or B frame, respectively.

s: the packet size (in bytes).

p: the packet loss probability.

t_{RTT}: the round-trip time (in milliseconds).

T: the capacity constraint (in packets per second), limited by the ISP or by a TCP-Friendly rate [9].

3.2 Distortion of Streaming MPEG

When a video is streaming over an unreliable network under a capacity constraint, its perceptual quality can be degraded by two factors: quantization and frame loss. The quantization distortion, caused by Quality Scaling which yields low accuracy of the DCT coefficients, appears visually as coarse granularity in every frame. Frame loss, caused by Temporal Scaling and network packet loss, appears visually as jerkiness in the video playout.

This section uses the Video Quality Model (VQM), an objective video quality measurement developed by the Institute for Telecommunication Sciences (ITS) [10], to measure the distortion from quantization. Section 3.3 uses the playable frame rate to capture the distortion from frame loss. Section 3.4 presents an overall measurement, namely, distorted playable frame rate, to combine these two factors.

VQM takes an original video and a distorted video as input and returns a distortion value D between 0 (no distortion) and 1 (highest distortion). Previous research shows the perceptual video distortion varies exponentially with the quantization value [3]. Our preliminary studies [14] measuring videos encoded with different quantization values with VQM show the distortion, D, can be approximated by an exponential function of the quantization value v_Q as:

$$D = \hat{D} \cdot {v_Q}^{\lambda_D} \tag{3}$$

where v_Q is the quantization value decided by l_{QS} as in Table 2, $\hat{D}$ is the VQM distortion when $v_Q = 1$, and λ_D is the exponential coefficient.

3.3 Playable Frame Rate

3.3.1 Frame Size

When the quantization values change, the frame sizes, and hence, streaming bitrate, change as well. Previous research shows the bitrate of an MPEG stream can be approximated by an exponential function of the quantization value v_Q [3]. Our preliminary experiments [14] suggest frame size can be estimated by an exponential function of quantization value v_Q, given as:

$$\begin{cases} S_I = \hat{S_I} \cdot {v_Q}^{\lambda_I} \\ S_P = \hat{S_P} \cdot {v_Q}^{\lambda_P} \\ S_B = \hat{S_B} \cdot {v_Q}^{\lambda_B} \end{cases} \tag{4}$$

where v_Q is the quantization value, $\hat{S_*}$ is the frame size when $v_Q = 1$, and λ_* is the exponential coefficient. Note, all the results S_* need to be rounded up to the nearest integer $\lceil S_* \rceil$ since each video frame must be divided into a whole number of packets when sent on the network.

3.3.2 Playable Frame Rate

Our previous work [15] derived a model to estimate total playable frame rate for streaming MPEG with Temporal Scaling. With the model, the total playable frame rate R is:

$$R = R(p, (N_{PD}, N_{BD}), (S_I, S_P, S_B), (S_{IF}, S_{PF}, S_{BF})) \tag{5}$$

Since N_{PD} and N_{BD} are decided by l_{TS} as in Table 1, and S_I, S_P, and S_B are decided by l_{QS} as in Equation 4 and Table 2, this equation can be written as:

$$R = R(p, l_{TS}, l_{QS}, (S_{IF}, S_{PF}, S_{BF})) \tag{6}$$

The estimated frame rate is then used to measure the quality distortion from frame loss, as indicated in Section 3.4.

3.4 Distorted Playable Frame Rate

When a video is streamed over a network with Quality Scaling and Temporal Scaling, the video quality distortion is determined by both the quantization and the frame loss. Encoding the video with a high quantization value makes every frame have a visually coarse granularity and yields intra-frame quality distortion. Any frame loss from Temporal Scaling and during transmission reduces the playable frame rate and results in visual jerkiness in the playout, yielding inter-frame quality distortion.

If the inter-frame and intra-frame distortion contribute independently to the overall distortion, then the video quality distortion can be represented by a function of these two factors. To produce the best quality video, the sender needs to use the best quantization level *and* the receiver needs to receive all the frames. So these two factors are combined in a multiplicative fashion, which is referred to as the distorted frame rate, R_D:

$$R_D = (1 - D) \cdot R \tag{7}$$

where D is the quality distortion from Equation 3 and R is the playable frame rate from Equation 6.

The motivation behind R_D is: if a video is streamed with the best quantization value, its intra-frame quality distortion is 0 and the video quality is decided only by the playable frame rate R. With any other quantization value, every frame carries less visual detail and its contribution to the video quality (measured by the frame rate) is reduced by the quality distortion D. A previous preliminary user study in [14] indicates a high correlation between user perceptual quality and distorted playable frame rate R_D, which suggests that R_D is effective in representing the overall video quality. A comprehensive user study in [13] confirms R_D can accurately estimate the user perceptual quality.

4. OPTIMIZATION ALGORITHM

For given network conditions and MPEG video parameters, the total distorted playable frame rate R_D varies with the Quality Scaling level, the Temporal Scaling level, and the amount of FEC for each type of frame as a function $R_D(p, l_{TS}, l_{QS}, (S_{IF}, S_{PF}, S_{BF}))$ where the streaming bitrate is limited by the capacity constraint, T. Thus, this model can be used to optimize the distorted playable frame rate, R_D, using the following operation research equation:

$$\begin{cases} Maximize: \\ \quad R_D = (1 - D(l_{QS})) \cdot R(p, l_{TS}, l_{QS}, (S_{IF}, S_{PF}, S_{BF})) \\ Subject\ to: \\ \quad G \cdot ((S_I(l_{QS}) + S_{IF}) + N_{PD}(l_{TS}) \cdot (S_P(l_{QS}) + S_{PF}) \\ \qquad + N_{BD}(l_{TS}) \cdot (S_B(l_{QS}) + S_{BF})) \leq T \end{cases} \tag{8}$$

Unfortunately, finding a closed form solution for the non-linear function R_D is difficult since there are many saddle points. However, given that the optimization problem is expressed in terms of integer variables over a restricted domain, an exhaustive search of the discrete space is feasible. With fixed input values for (p, RTT, s), G and functions of $(N_{PD}(l_{TS}), N_{BD}(l_{TS})$ and $(S_I(l_{QS}), S_P(l_{QS}), S_B(l_{QS}))$, each set of values of l_{TS}, l_{QS}, and (S_{IF}, S_{PF}, S_{BF}) can determine the distorted playable frame rate R_D as follows:

1. l_{QS} is used to obtain a quantization value v_Q from Table 2. The video frame sizes (S_I, S_P and S_B) are then approximated using Equation 4.

2. The video streaming bitrate is estimated using the video frame sizes, the FEC frame sizes and (N_{PD}, N_{BD}). If the estimated bitrate is larger than the capacity constraint T, the set of values of l_{QS}, l_{TS} and (S_{IF}, S_{PF}, S_{BF}) are invalid and R_D is returned as 0.

3. Otherwise, the playable frame rate R is estimated by inputting (p, l_{TS}, l_{QS}, S_{IF}, S_{PF}, S_{BF}) into Equation 5.

4. Using v_Q, the distortion from quality scaling D is approximated using Equation 3.

5. Knowing R and D, the distorted playable frame rate R_D is estimated using Equation 7.

With these steps for each set of values, the space of possible values for l_{TS}, l_{QS} and (S_{IF}, S_{PF}, S_{BF}) can be exhaustively searched to determine the scaling levels and the amount of FEC packets for each frame type to yield the maximum distorted playable frame rate under the capacity constraint. In fact, the computation required by the search can be done in real-time, making the determination of optimal choices for adaptive FEC feasible for streaming MPEG.

5. ANALYSIS

5.1 Methodology

Using the optimization algorithm presented in Section 4, the distorted playable frame rates over a range of network and application settings are explored. For each set of network and application parameters, the distorted playable frame rates are compared for MPEG streaming with Temporal Scaling, Quality Scaling and the combination of them.

The MPEG streams with scaling are protected by one of four different FEC methods:

1. Adjusted FEC: Before transmitting, the sender employs the optimization algorithm to determine the FEC and scaling levels that produce the maximum distorted playable frame rate R_D and uses these for the entire video transmission.

2. Large Fixed FEC: The sender protects each frame with FEC packets equivalent to 15% of the original frame size (moved up to the nearest integer). This FEC pattern provides strong protection to each frame and roughly represents the relative importance of the I, P and B frames [4].

3. Small Fixed FEC: Each I frame receives 1 FEC packet. This simple FEC pattern protects the most important frame, the I frame [1].

4. Non-FEC: The sender adds no FEC to the video.

5.2 System Settings

A commonly-used MPEG GOP pattern, '`IBBPBBPBBPBBPBB`', and a typical full motion frame rate of 30 frames per second (fps) are used. The packet size s of 1 KB, round-trip time t_{RTT} of 50ms and packet loss probability p, which ranges from 0.005 to 0.08 in steps of 0.005, are chosen based on the characteristics of many network connections [5].

Motion	Video	Description
Low	Container	A working container ship
Low	Hall	A hallway
Low	News	Two news reporters
Medium	Foreman	A talking foreman
Medium	Paris	Two people talking with high-motion gestures
Medium	Silent	A person demonstrating sign language
High	Coastguard	Panning of a moving coastguard ship
High	Mobile	Panning of moving toys
High	Vectra	Panning of a moving car

Table 4: Video Clips

Nine video clips are used for the experiments. Each video clip has 300 raw images and the size of each frame is 352x288 pixels (CIF). Table 4 provides an approximate motion classification of each video clip, with an identifying name and a short description of the video content.

5.3 Comparison of Scaling methods

Figure 1 depicts the distorted playable frame rates for the three scaling methods with Adjusted FEC for the nine videos. The x-axes (bottom) are the packet loss probabilities, the x2-axes (top) depict the corresponding TCP-Friendly capacity constraints, and the y-axes are the distorted playable frame rates. From the data in these figures, the combination of Temporal and Quality Scaling provides the best quality under all network and video conditions. When the packet loss is low and capacity limit is high, Quality Scaling provides performance nearly the same as the combination of Temporal and Quality scaling. However, when the loss rate is high and the capacity is limited, Quality Scaling cannot scale enough to reduce the streaming bitrate below the capacity constraint and must be used with Temporal Scaling to yield reasonable video quality. These trends hold for all videos despite the differences in content among the clips. However, the motion properties of video clips are correlated to the differences between Temporal Scaling and Quality Scaling. For the high motion videos

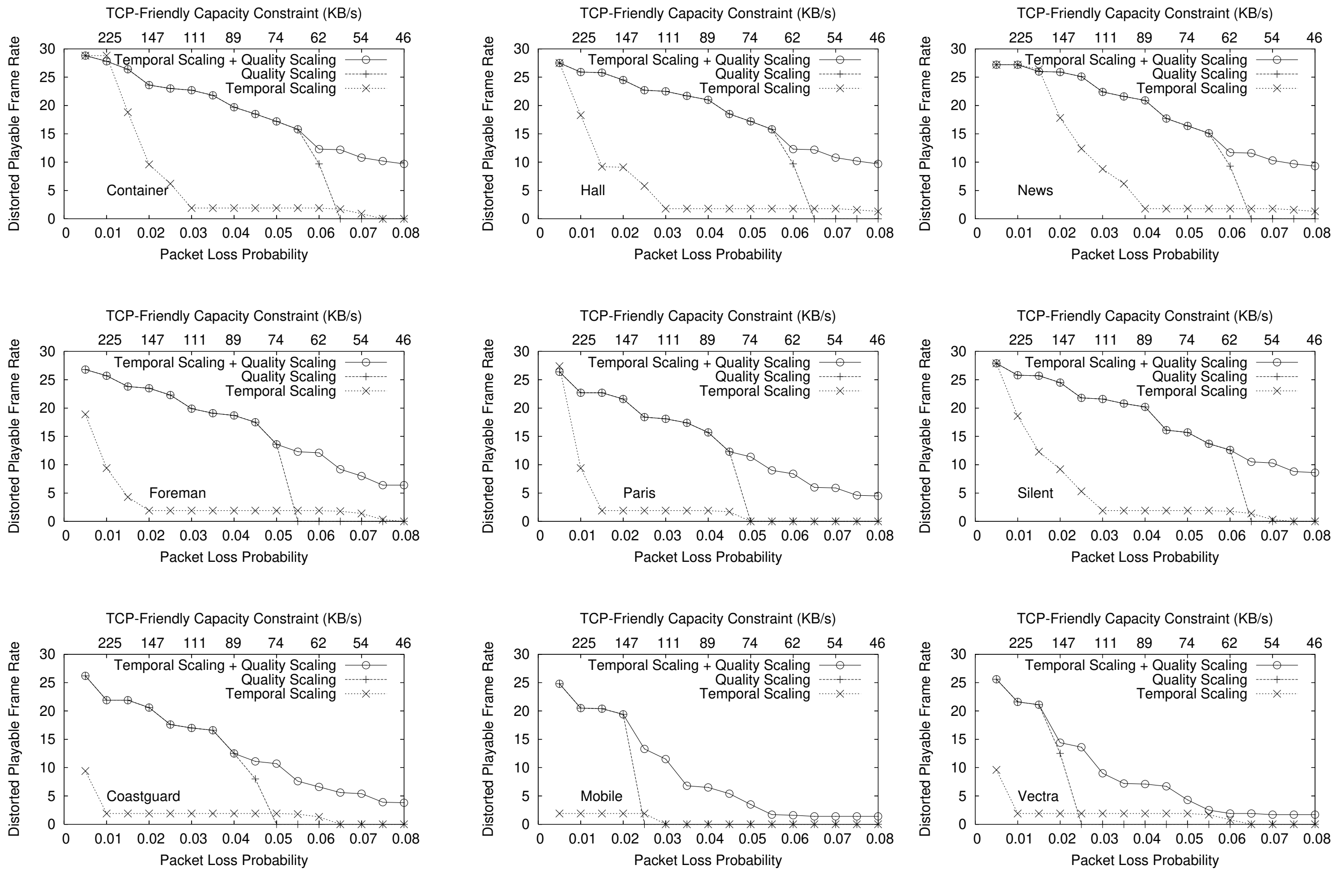

Figure 1: Comparison of scaling methods, with the 1st row of low motion clips, the 2nd row of medium motion clips and the 3rd row of high motion clips

(*Coastguard*, *Mobile* and *Vectra*), Quality Scaling is always much better than Temporal Scaling. For the low motion videos (*Container*, *Hall* and *News*), Temporal Scaling can also provide reasonably high quality when the loss rate is low. The video motion also decides the point at which Temporal Scaling should be combined with Quality Scaling to provide a reasonable video quality, with high-motion videos needing the combination for lower capacity constraints than do low-motion videos.

5.4 Comparison of FEC Methods

Figure 2 depicts the distorted playable frame rates for the four FEC choices with the combination of Temporal Scaling and Quality Scaling for the nine videos. The x-axes (bottom) are the packet loss probabilities, the x2-axes (top) depict the corresponding TCP-Friendly capacity constraints, and the y-axes are the distorted playable frame rates. From the data in these figures, Adjusted FEC provides the best quality under all network and video combinations. The benefits in quality for Adjusted FEC over Non-FEC are substantial, with Adjusted FEC providing 5-10 more frames per second for all loss rates. The Small Fixed FEC approach usually improves playable frame rates over Non-FEC, especially when loss rates are high. However, Small Fixed FEC yields frame rates that are still much lower than the frame rates with Adjusted FEC. The Large Fixed FEC approach achieves the playable frame rate provided by Adjusted FEC at low loss rates since the TCP-Friendly bitrate is relatively high. However, when the capacity is limited, Large Fixed FEC requires too much overhead and results in less video data being sent. These trends hold for all the videos despite the differences in motion content among the clips.

6. CONCLUSION

This paper systematically compares Temporal Scaling, Quality Scaling and their combination, for streaming MPEG videos over a range of network and video content conditions. An analytic model is proposed that captures the quality distortion of streaming MPEG in the presences of Temporal Scaling, Quality Scaling and repair with Forward Error Correction footnoteNotice, although other error correction methods [12] can be used to repair loss, our results are intended to show that dynamic adjusted repair performs better than static repair, especially when there are packet loss and capacity constraints., as well as network and video parameters. Using this model with an optimization algorithm determines the optimal FEC and scaling given a capacity bound.

Analytic experiments on nine videos with varied motion characteristics show that when capacity constraints are moderate and loss rates are low, Temporal Scaling adds very little to the quality produced by using only Quality Scaling. When bitrates are low and loss rates are high, the combination of Quality Scaling and Temporal Scaling can still pro-

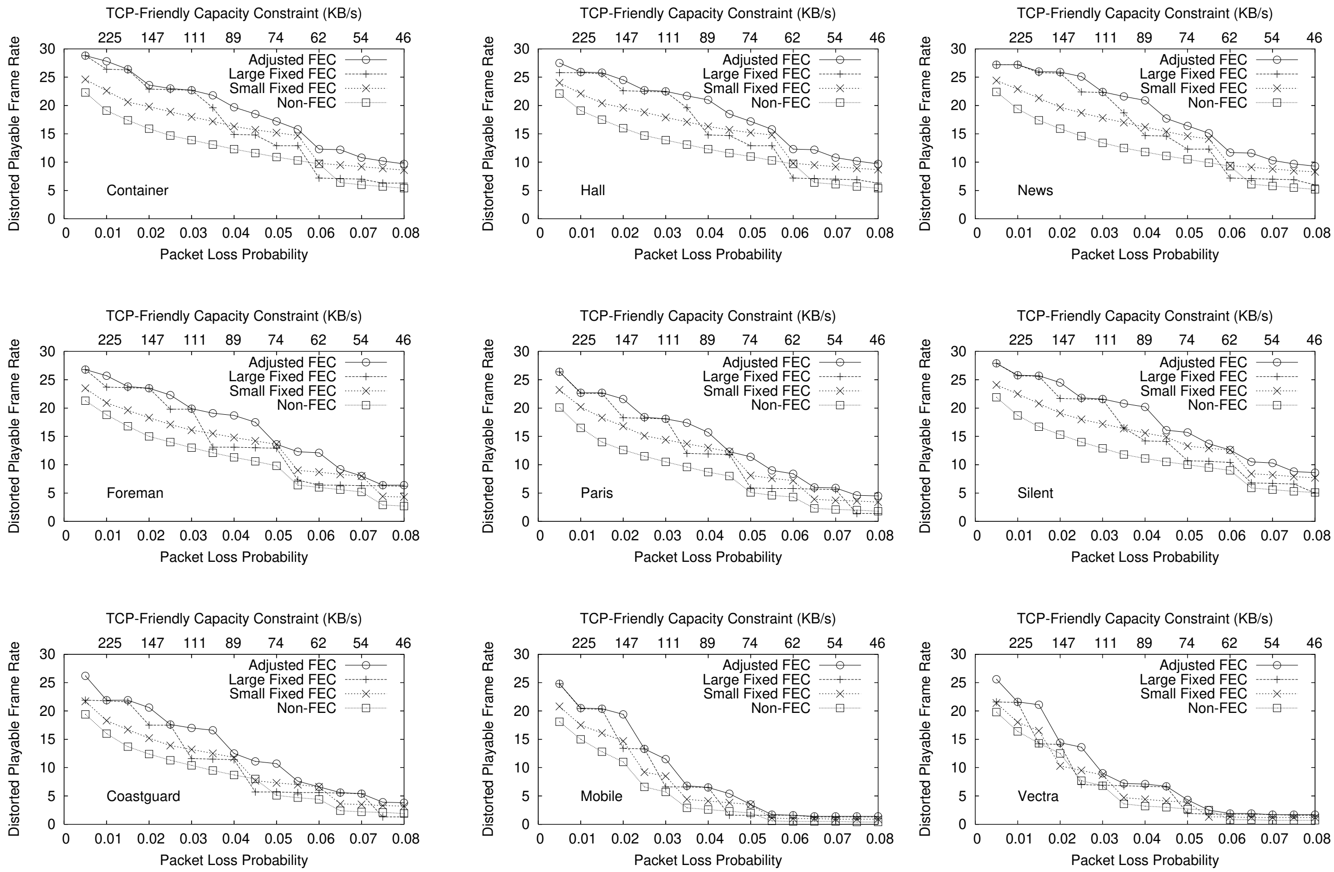

Figure 2: Comparison of FEC methods, with the 1st row of low motion clips, the 2nd row of medium motion clips and the 3rd row of high motion clips

vide reasonable video quality. Additionally, the results imply that Quality Scaling provides better quality video than Temporal Scaling and that differences in their performance is correlated to video motion characteristics. Under conditions with loss, Adjusted FEC always achieves higher quality than MPEG video with any Fixed FEC or without FEC.

7. REFERENCES

[1] N. Feamster and H. Balakrishnan. Packet Loss Recovery for Streaming Video. In *Proceedings of Packet Video Workshop (PV)*, Pittsburgh, PA, USA, Apr. 2002.

[2] S. Floyd and K. Fall. Promoting the Use of End-to-End Congestion Control in the Internet. *IEEE/ACM Transactions on Networking*, Feb. 1999.

[3] P. Frossard and O. Verscheure. Joint Source/FEC Rate Selection for Quality-Optimal MPEG-2 Video Delivery. *IEEE Transactions on Image Processing*, 10(12):1815–1825, Dec. 2001.

[4] F. Hartanto and H. R. Sirisena. Hybrid Error Control Mechanism for Video Transmission in the Wireless IP Networks. In *Proceedings of IEEE LANMAN*, Nov. 1999.

[5] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley. Inferring TCP Connection Characteristics Through Passive Measurements. In *Proceedings of IEEE Infocom*, Hong Kong, China, Apr. 2004.

[6] C. Krasic, J. Walpole, and W.-C. Feng. Quality-adaptive Media Streaming by Priority Drop. In *Proceedings of NOSSDAV*, Monterey, CA, USA, June 2003.

[7] K. Mayer-Patel, L. Le, and G. Carle. An MPEG Performance Model and Its Application To Adaptive Forward Error Correction. In *Proceedings of ACM Multimedia*, December 2002.

[8] J. McCarthy, M. A. Sasse, and D. Miras. Sharp or Smooth? Comparing the Effects of Quantization versus Frame Rate for Streamed Video. In *Proceedings of ACM CHI Conference on Human Factors in Computing Systems*, Vienna, Austria, Apr. 2004.

[9] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and Its Empirical Validation. In *Proceedings of ACM SIGCOMM*, Vancouver, Brisish Columbia, Candada, 1998.

[10] M. Pinson and S. Wolf. A New Standardized Method for Objectively Measuring Video Quality. *IEEE Transactions on Broadcasting*, 50(3):312–322, Sept. 2004.

[11] I. S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *Journal of the Society of Industrial and Applied Mathematics (SIAM)*, 8(2):300–304, June 1960.

[12] X. Su and B. W. Wah. Multi-Description Video Streaming with Optimized Reconstruction-Based DCT and Neural-Network Compensations. *IEEE Transactions on Multimedia*, 3(1):123–131, Mar. 2001.

[13] H. Wu, M. Claypool, and R. Kinicki. ARMOR - A System for Adjusting Repair and Media Scaling for Video Streaming.

[14] H. Wu, M. Claypool, and R. Kinicki. Adjusting Forward Error Correction with Quality Scaling for Streaming MPEG. In *Proceedings of NOSSDAV*, Stevenson, Washington, USA, June 2005.

[15] H. Wu, M. Claypool, and R. Kinicki. Adjusting Forward Error Correction with Temporal Scaling for TCP-Friendly Streaming MPEG. *ACM TOMCCAP*, 1(4), Nov. 2005.