Adjusting Forward Error Correction with Temporal Scaling for TCP-Friendly Streaming MPEG

HUAHUI WU, MARK CLAYPOOL, and ROBERT KINICKI
Worcester Polytechnic Institute

New TCP-friendly constraints require multimedia flows to reduce their data rates under packet loss to that of a conformant TCP flow. To reduce data rates while preserving real-time playout, temporal scaling can be used to discard the encoded multimedia frames that have the least impact on perceived video quality. To limit the impact of lost packets, Forward Error Correction (FEC) can be used to repair frames damaged by packet loss. However, adding FEC requires further reduction of multimedia data, making the decision of how much FEC to use of critical importance. Current approaches use either inflexible FEC patterns or adapt to packet loss on the network without regard to TCP-friendly data rate constraints. In this article, we analytically model the playable frame rate of a TCP-friendly MPEG stream with FEC and temporal scaling, capturing the impact of distributing FEC within MPEG frame types with interframe dependencies. For a given network condition and MPEG video encoding, we use our model to exhaustively search for the optimal combination of FEC and temporal scaling that yields the highest playable frame rate within TCP-friendly constraints. Analytic experiments over a range of network and application conditions indicate that adjustable FEC with temporal scaling can provide a significant performance improvement over current approaches. Extensive simulation experiments based on Internet traces show that our model can be effective as part of a streaming protocol that chooses FEC and temporal scaling patterns that meet dynamically-changing application and network conditions.

Categories and Subject Descriptors: C.2.m [**Computer-Communication Networks**]: Miscellaneous

General Terms: Performance, Design

Additional Key Words and Phrases: Multimedia networking, MPEG, forward error correction, TCP-friendly

1. INTRODUCTION

As the number of active Internet users continues to grow and streaming media applications become more commonplace, the number of flows and the volume of data traversing the Internet is increasing quickly. The sheer number of possible users and applications at any point in time raises the probability of streaming multimedia flows encountering congestion problems. To overcome short-term congestion and avoid long-term congestion collapse, the Internet relies upon the congestion control mechanisms in Transmission Control Protocol (TCP), the current dominant transport protocol on the Internet.

While streaming flows have traditionally selected UDP over TCP [Mena and Heidemann 2000; Wang et al. 2001], there is a growing consensus that all Internet applications must be TCP-friendly. A flow is *TCP-friendly* if its data rate does not exceed the maximum data rate from a conformant TCP connection under equivalent network conditions. There are proposed approaches to detect and restrict the bandwidth of non-TCP-friendly flows [Mahajan et al. 2001]. Thus, networking researchers have

Authors' address: H. Wu, M. Claypool and R. Kinicki, Computer Science Department, Worcester Polytechnic Institute, 100 Institute Rd, Worcester, MA 01609; email: {flashine,claypool,rek} @cs.wpi.edu.

proposed new TCP-friendly protocols (e.g., TFRC) [Balakrishnan et al. 1999; Floyd et al. 2000; Rejaie et al. 1999] for transporting streaming multimedia. By requiring TCP-friendly streaming protocols, the belief is that router Active Queue Management techniques can more effectively respond to all forms of congestion. This, in turn, should yield better overall quality of service for streaming flows.

To preserve real-time streaming media playout, multimedia servers must scale back their streaming data rate to match the TCP-friendly data rate. This proactive data rate reduction by the multimedia server is called *media scaling* [Bocheck et al. 1999; Tripathi and Claypool 2002]. *Temporal scaling* is a widely used form of media scaling whereby the multimedia server selectively discards frames prior to transmission. Armed with knowledge about the relative importance of specific frame types and interframe dependencies, a multimedia application can discard the least significant packets with respect to perceived quality, while a congested router can only randomly drop packets [Hemy et al. 1999]. While proposed congestion marking schemes such as Early Congestion Notification (ECN) [Floyd 1994] can reduce packet loss dramatically by having congested routers mark packets instead of dropping them, ECN has not seen widespread deployment in the ten years since it has been proposed [DeSantis and Loose 2003], making ECN's future uncertain. Moreover, even with congestion marking, under severe congestion or during channel errors, multimedia streams will still experience packet loss.

While multimedia applications can tolerate some data loss, excessive packet loss during congestion yields unacceptable media quality. Since video encoding involves interframe dependencies [Mitchell and Pennebaker 1996], the random dropping of packets by routers can seriously degrade video quality. In MPEG, for example, dropping packets from an independently encoded I frame causes the following dependent P and B frames to not be fully decodable. In practice, interframe dependencies convert a 3% packet loss rate into a 30% frame loss rate [Boyce and Gaglianello 1998].

Although TCP can successfully recover from packet losses using retransmissions, videoconferencing and interactive virtual reality applications cannot afford to use retransmission mechanisms when round-trip times for the streaming flow are high. While current Internet backbone routers are able to reduce queuing delays to near zero [Boutremans et al. 2002], recent measurement [Jaiswal et al. 2004] indicates that 30% of all flows have a median round-trip time above 600 milliseconds. This makes retransmissions impractical for these multimedia applications if interactive delay bounds are to be observed.[1] This suggests utilizing lower latency repair approaches, such as Forward Error Correction (FEC), in conjunction with TCP-friendly protocols to deliver streaming applications over the Internet. Used properly, FEC [Bolot et al. 1999; Nguyen and Zakhor 2002; Padhye et al. 2000; Park and Wang 1999] can reduce or eliminate packet loss and partially or fully insulate video applications from degraded quality [Liu and Claypool 2000]. However, FEC requires additional repair data to be added to the original video data. If a streaming video is to operate within TCP-friendly bandwidth limits, the additional FEC data implies a lower effective transmission rate for the original video content.

Assuming the desirability of a TCP-friendly multimedia protocol and the availability of an estimate of the current packet loss rate along a flow path, selecting the best distribution of FEC packets within video frames with inherent interframe encoding dependencies can be cast as a constrained optimization problem that attempts to optimize the quality of the video stream [Mayer-Patel et al. 2002]. Current approaches use either a priori, static FEC [Albanese et al. 1996; Hardman et al. 1995] or adapt FEC to perceived packet loss on the network without regard to TCP-friendly data rate constraints [Bolot et al. 1999; Padhye et al. 2000; Park and Wang 1999].

In Feamster and Balakrishnan [2002], the authors derived a relationship between the packet loss rate and the observed frame rate, but they did not model repair or media scaling. Previously, we derived

[1]The International Telecommunication Union states that one-way delays of over 300 milliseconds result in poor quality for interactive audio applications [1996].

an analytic model of MPEG frame dependencies and FEC to compute an achievable frame rate in the presence of packet loss [Wu et al. 2003a]. Compared to related work [Mayer-Patel et al. 2002], our previous model more accurately captures the dependencies of P frames and uses integer parameters to reduce search time and improve efficiency. Building to our previous work [Wu et al. 2003a], this article adds two important contributions. First, our earlier model did not account for temporal scaling which when used in practice results in a slow motion playout of the video. This article incorporates a model for temporal scaling to adjust the streaming bitrate that preserves real-time video playout in the face of network capacity constraints. Second, while our earlier work included only evaluation using analytic modeling, this article provides a comparison of the new analytic model to simulations based on Internet measurements and traces.

The enhanced model characterizes the performance of temporally-scaled MPEG video with Forward Error Correction in the presence of packet loss. Assuming the network protocol provides loss rates, round-trip times, and packet sizes, and the streaming video application provides details on the MPEG frame sizes and types, the model allows specification of the number of FEC packets per MPEG frame type and the temporal scaling pattern and computes the total playable frame rate. Since the two main optimizations afforded by our model (temporal scaling and FEC) determine the number of playable frames at the receiver, frame rate is used as the measure of performance. While alternate performance measures, such as peak signal-to-noise ratio (PSNR) or the video quality metric (VQM) [Pinson and Wolf 2004], may be more appropriate when quality scaling is used to reduce video bitrates, quality scaling and related performance metrics are beyond the scope of this work. We use our model to exhaustively search all possible combinations of FEC and temporal scaling patterns to find the combination of FEC and temporal scaling that yields the maximum playable frame rate under the TCP-friendly bandwidth constraint. The analytic calculations required by the search can be done in real-time, making the determination of optimal choices for adaptive FEC feasible for most streaming multimedia connections.

Since the optimal solution from the analytic model for adjusted FEC depends on accurate estimates of packet loss and round-trip time and upon fixed MPEG frame sizes, simulation experiments are designed that explore the effectiveness of using our model under realistic Internet conditions. The experimental results demonstrate that even with 100% error in the estimated packet loss probability and bursty packet loss, using our model to adjust FEC and temporal scaling pattern provides predictions within 1.8 frames per second of the actual playable frame rate. Additionally, since the analytic model assumes constant round-trip time and fixed MPEG frame sizes, we constructed additional simulation experiments with trace-driven round-trip times and MPEG frame sizes. These simulated results imply that the analytic model does a good job of selecting the FEC distribution for the video stream despite using only an average round-trip time and a fixed MPEG frame size for each frame type. The cumulative effect of the experiments presented is to lend credence to using the enhanced model to effectively adjust FEC with temporal scaling to provide high playable frame rates for TCP-friendly streaming video.

The remainder of the article is organized as follows. Section 2 provides background knowledge and clarifies terminology; Section 3 introduces the analytic model for adjustable FEC; Section 4 presents analytic experiments using our model; Section 5 presents simulation experiments that show the feasibility of using our model under realistic network conditions; and Section 6 summarizes the article and presents possible future work.

2. BACKGROUND

This section provides background and clarifies terminology on TCP-friendliness, forward error correction, MPEG video and temporal scaling to facilitate the development of the analytic model introduced in the next section.

2.1 TCP-Friendly Flows

A flow is considered to be *TCP-friendly* if its bandwidth usage in steady-state is no more than an equivalent conformant TCP flow running under comparable network conditions (i.e., packet drop rate and round-trip time). Padhye et al. [1998] analytically derived the following equation for TCP throughput:

$$T = \frac{s}{t_{RTT}\sqrt{\frac{2p}{3}} + t_{RTO}\left(3\sqrt{\frac{3p}{8}}\right)p(1+32p^2)}, \quad (1)$$

where s is the packet size, t_{RTT} is the round-trip time, p is the steady-state packet loss probability, t_{RTO} is the TCP retransmit timeout value.[2] Thus, Equation (1) provides an upper bound, T, for the TCP-friendly sending rate. Flows that are not TCP-friendly can seize a disproportionate share of the network's capacity. Besides being unfair, this type of unresponsive behavior by numerous streaming flows may lead to Internet congestion collapse [Braden et al. 1998; Floyd and Fall 1999]. Thus, for the Internet to support the future demands for multimedia applications, this research assumes transport protocols such as Balakrishnan et al. [1999], Floyd et al. [2000], and Rejaie et al. [1999] that can keep multimedia streaming flows TCP-friendly.

2.2 Forward Error Correction (FEC)

Streaming video frames are often larger than a single Internet packet. Since Internet congestion results in lost packets, we apply FEC at the packet level. Thus, we model an application-level video frame as being transmitted in K packets where K varies with frame type, encoding method, and media content. Media independent FEC [Reed and Solomon 1960] consists of adding $(N-K)$ redundant packets to the K original packets and sending the N packets as the frame. If any K or more packets are successfully received, the frame can be completely reconstructed. Although the additional delay needed to create redundant FEC packets cannot be ignored given application delay constraints, Rizzo [1997] shows that software FEC can be done in real-time with data rates up to 100 Mbps. If necessary, hardware can be used to speed up FEC encoding even more.

To analyze the effects of FEC on video frames, we model the sending of packets 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:

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

Since Equation (2) ignores the bursty nature of Internet packet losses, we evaluate the impact of this simplifying assumption in Section 5.

2.3 MPEG

The MPEG[3] standard is gaining in popularity and appears a viable open standard for video on the Internet [Mitchell and Pennebaker 1996]. MPEG uses both intraframe and interframe compression. I (intracoded) frames are encoded independently of other frames and focus on encoding similarities within a video scene. P (predictive-coded) frames are encoded based on motion differences from preceding I or P frames in the video sequence. B (bidirectionally predictive-coded) frames are encoded based on motion differences from preceding and succeeding I or P frames.

[2]We set t_{RTO} to be $4 \times t_{RTT}$ as in Floyd et al. [2000].
[3]Motion Picture Expert Group, available at http://www.chiariglione.org/mpeg/.

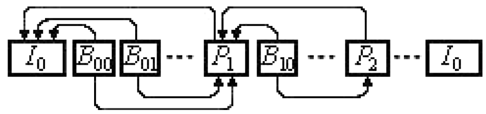


Fig. 1. A sample MPEG Group of Pictures (GOP).

MPEG video typically repeats a pattern of I, P, and B frames (known as a Group of Pictures or GOP) for the duration of a video stream. Figure 1 shows a sample GOP where the second I frame in the figure marks the beginning of the next GOP, and the arrows indicate frame dependency relationships. Because of the dependencies of the I, P, and B frames, the loss of one P frame can severely degrade the quality of other P and B frames, and the loss of one I frame can impact the quality of the entire GOP. This implies that I frames are more important than P frames, and P frames are more important than B frames.

Since B frames cannot be decoded until the subsequent I or P frame has arrived,[4] B frames introduce an additional playout delay of one or more interframe times. However, this added delay can be controlled by limiting the number of B frames in a row. For example, two B frames in a row, a number typical of many GOPs, in a video encoded at 30 frames per second introduces an additional delay of only 66 milliseconds. This article assumes this added delay is tolerable compared to delays induced by the network. However, even in the event that all B frames are discarded, the MPEG model presented in this article is still valid.

Let N_P represent the number of P frames in a GOP, N_B represent the number of B frames in a GOP, and N_{BP} represent the number of B frames in between an I and a P frame or two P frames.[5] Thus, $N_B = (1 + N_P) \times N_{BP}$. Using this notation, a GOP pattern can be uniquely identified by GOP(N_P,N_B). For example, GOP(3,8) indicates the GOP pattern `'IBBPBBPBBPBB'`. Unless specifically indicated, GOP(3,8), a commonly used pattern on the Internet [Acharya and Smith 1998], is used for the remainder of this article as the fixed GOP pattern. While we have studied the potential for longer GOPs to result in higher playable frame rates, the benefits are quite marginal[6] and can result in propagation of errors if original references are used during the encoding of P frames. Analysis of other GOP patterns can be found in Wu et al. [2003b].

We use the subscripting notation presented in Figure 1 to identify individual frames within a GOP. The single I frame of a GOP is referred to as I_0, while P frames are P_i, where $1 \leq i \leq N_P$, and B frames are B_{ij}, where $0 \leq i \leq N_P$ and $0 \leq j < N_{BP}$. For example, P_3 is the third P frame, and B_{01} is the second B frame in the first interval of I and P frames.

2.4 Temporal Scaling

To preserve the timing aspects of real-time streaming video, the application data rate must be adjusted to the available network bitrate (i.e., the TCP-friendly rate). This is commonly done by *temporal scaling* in which lower priority video frames are discarded prior to the GOP transmission. For instance, with the GOP(3,8) pattern of "`IBBPBBPBBPBB`", the data rate can be approximately halved by discarding all the B frames and only sending "`I--P--P--P--`".

We use N_{PD} to denote the number of P frames sent in one GOP, and N_{BD} to denote the number of B frames delivered in one GOP ($N_P - N_{PD}$ P frames are then discarded, and $N_B - N_{BD}$ B frames are discarded). For instance, if temporal scaling of GOP(3,8) results in "`I--P--P--P--`" being sent, then N_{PD} is three and N_{BD} is 0. To clarify the temporal scaling decision, we introduce a binary coefficient $D_\#$ (e.g.,

[4]In fact, the following I or P frame is often transmitted before the dependent B frame for this reason.

[5]As in typical MPEG videos, we assume B frames are distributed evenly in the intervals between I and P frames.

[6]GOP(20,42) provides only one fps higher than GOP(3,8) under many network conditions [Wu et al. 2004].

Table I. Temporal Scaling Characteristics

Scaling Level	N_{PD} $0 \sim 3$	N_{BD} $0 \sim 8$	Scaling Pattern	Binary Coefficient $D_{\#}$											
				I	B_{00}	B_{01}	P_1	B_{10}	B_{11}	P_2	B_{20}	B_{21}	P_3	B_{30}	B_{31}
0	3	8	`IBBPBBPBBPBB`	1	1	1	1	1	1	1	1	1	1	1	1
1	3	7	`IBBPBBPBBPB-`	1	1	1	1	1	1	1	1	1	1	1	0
2	3	6	`IBBPBBPB-PB-`	1	1	1	1	1	1	1	1	0	1	1	0
3	3	5	`IBBPB-PB-PB-`	1	1	1	1	1	0	1	1	0	1	1	0
4	3	4	`IB-PB-PB-PB-`	1	1	0	1	1	0	1	1	0	1	1	0
5	3	3	`IB-PB-PB-P--`	1	1	0	1	1	0	1	1	0	1	0	0
6	3	2	`IB-PB-P--P--`	1	1	0	1	1	0	1	0	0	1	0	0
7	3	1	`IB-P--P--P--`	1	1	0	1	0	0	1	0	0	1	0	0
8	3	0	`I--P--P--P--`	1	0	0	1	0	0	1	0	0	1	0	0
9	2	0	`I--P--P-----`	1	0	0	1	0	0	1	0	0	0	0	0
10	1	0	`I--P--------`	1	0	0	1	0	0	0	0	0	0	0	0
11	0	0	`I-----------`	1	0	0	0	0	0	0	0	0	0	0	0

D_I, D_{P_2}, or $D_{B_{11}}$) where # can be replaced by I or P or B frame. Specifically, $D_{\#}$ is 0 if temporal scaling discards frame # prior to GOP transmission, and $D_{\#}$ is 1 if frame # will be sent.

While temporal scaling could, in theory, select any of the frames in a GOP to discard, the following set of strategies take into account MPEG frame dependencies and minimizes the effect of temporal scaling on the quality of the received video.

(1) Since B frames depend on I and P frames, B frames are discarded evenly before discarding I or P frame.

(2) Since each P frame depends upon the previous P frame or I frame, P frames are discarded from the back (last) to the front of the GOP pattern.

(3) Since every frame in a GOP depends upon the I frame directly or indirectly, I frames are never discarded.

Table I lists all the possible temporal scaling levels for GOP(3,8) with these rules. Each line tells the values of N_{PD} and N_{BD} as well as the scaling patterns and the binary coefficients for that scaling level.

3. ANALYTICAL MODEL

This section develops the analytic model used to determine the playable frame rate of TCP-friendly streaming video flows with adjusted FEC and temporal scaling in the presence of network packet loss. First, we identify application and network parameters related to TCP-friendly MPEG streams (see Section 3.1). Next, working from MPEG frame sizes and adjustable amounts of FEC per frame type, we create a system of equations to characterize the probability of successful transmission and playout for each MPEG frame type (see Section 3.2). We then incorporate temporal scaling and MPEG frame dependencies and derive formulas for transmission rate and playable frame rate (see Section 3.3). Lastly, considering a TCP-friendly bandwidth constraint, we optimize the playable frame rate by adjusting the temporal scaling and amount of FEC per frame (see Section 3.4).

3.1 Software Layers and Parameters

In our model, we incorporate the software layers and parameters indicated in Table II, where the parameters are:

—R_F: the maximum playable frame rate achieved when there is enough available capacity and no loss (typical full-motion video rates have $R_F = 30fps$);

Table II. Software Layers and Parameters

Layer	Parameters
MPEG	$S_I, S_P, S_B, N_P, N_B, R_F$
AFEC	$S_{IF}, S_{PF}, S_{BF}, N_{PD}, N_{BD}$
Network	p, t_{RTT}, s

—S_I, S_P, S_B: the number of packets for each I, P, or B frame, respectively;

—N_P, N_B: the number of P or B frames in one GOP, respectively;

—N_{PD}, N_{BD}: the number of P or B frames, respectively, sent per GOP after temporal scaling;

—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);

For a streaming session, we assume the network protocol provides loss rates, round-trip times, and packet sizes, while the streaming video application provides details on the MPEG frame characteristics. The model we develop in the rest of this section allows exploration of the effects various choices of FEC and temporal scaling have on application performance. In particular, we assume an AFEC (Adaptable FEC) component within the streaming application that adjusts the FEC and temporal scaling patterns so as to optimize the total playable frame rate.

3.2 Successful Frame Transmission Probabilities

Given I, P, and B frame sizes and the distribution of redundant FEC packets added to each frame type, the following equation provides the probability of successful transmission for each frame type:

$$\begin{aligned} q_I &= q(S_I + S_{IF}, S_I, p) \\ q_P &= q(S_P + S_{PF}, S_P, p) \\ q_B &= q(S_B + S_{BF}, S_B, p) \end{aligned} \tag{3}$$

where $q(.)$ defines the successfully transmission probability as an independent Bernoulli trial as in Equation (2), S_I. S_P, and S_B are the frame sizes; S_{IF}, S_{PF}, and S_{BF} are the FEC amounts in packets for I, P, and B frames; and p is the packet loss rate.

3.3 Playable Frame Rate

First, our model expresses the GOP rate (GOPs per second) analytically (see Section 3.3.1). Subsequently, the model computes the playable frame rate using the frame dependency relationships for each of the I, P, and B frame types (see Sections 3.3.2–3.3.4). Summing the individual playable frame rates provides the total playable frame rate for the streaming application (see Section 3.3.5).

3.3.1 *GOP Rate.* If, in adapting to the current available network bitrate, the GOP rate is decreased, the video will appear to run in slow motion. Thus, the GOP rate, G, must be kept constant in order to maintain the real-time playout speed at the receiver. Given R_F, the target full-motion frame rate, the GOP rate (specified in GOPs per second during encoding) is:

$$G = \frac{R_F}{(1 + N_P + N_B)}. \tag{4}$$

Temporal scaling is used to adapt the bitrate to the current available network capacity by discarding frames before transmission. This implies the ability to maintain a constant GOP rate.

3.3.2 *Playable Rate of I Frames.* Since I frames are independently encoded, the playable rate of I frames is simply the number of I frames transmitted successfully over the network

$$R_I = G \cdot q_I \cdot D_I, \tag{5}$$

where D_I is the binary coefficient which indicates if this I frame should be dropped for temporal scaling as in Section 2.4.

Since losing the I frame impacts the decodability of all subsequent frames in the GOP, this article fixes D_I to 1. Hence, $R_I = G \cdot q_I$.

3.3.3 *Playable Rate of P Frames.* The first P frame, P_1, can only be displayed when its preceding I frame and itself are successfully transmitted. Thus, P_1's playable frame rate is $R_{P_1} = R_I \cdot q_P \cdot D_{P_1}$, where D_{P_1} is the binary coefficient which indicates if this P frame should be dropped for temporal scaling as in Section 2.4. Since each subsequent P_i in the GOP depends upon the success of P_{i-1} and its own successful transmission, we have:

$$R_{P_i} = R_I \cdot {q_P}^i \cdot \prod_{k=1}^{i} D_{P_k}. \tag{6}$$

Using the temporal scaling strategies in Section 2.4, P frames are discarded back to front in the GOP and the playable rate of P frames is

$$R_P = \sum_{i=1}^{N_{PD}} R_{P_i} = G \cdot q_I \cdot \frac{q_P - q_P^{1+N_{PD}}}{1 - q_P}. \tag{7}$$

3.3.4 *Playable Rate of B Frames.* All N_{BP} adjacent B frames have the same dependency relationship (they depend upon the previous and subsequent I or P frame) and thus these B frames all have the same playable rate.

When a B frame precedes a P frame, the B frame depends only on that P frame. It is not necessary to consider the I or P frames before this P frame since these dependency relationships have already been accounted for in the successful transmission probability of the P frame. Thus,

$$R_{B_{ij}} = R_{P_{i+1}} \cdot q_B \cdot D_{B_{ij}} \quad when \quad 0 \le i \le N_P - 1, \tag{8}$$

where $D_{B_{ij}}$ is the binary coefficient which indicates if this B frame should be dropped for temporal scaling as in Section 2.4.

When a B frame precedes an I frame, the B frame depends upon both the preceding P frame and upon the succeeding I frame. For these B frames:

$$R_{B_{ij}} = R_{P_i} \cdot q_B \cdot D_{B_{ij}} \cdot q_I \quad when \quad i = N_P. \tag{9}$$

Finally, the playable rate for all B frames is:

$$R_B = \textstyle\sum_{i=0}^{N_P} \sum_{j=0}^{N_{BP}} R_{B_{ij}}. \tag{10}$$

3.3.5 *Total Playable Frame Rate.* The total playable frame rate is the sum of the playable frame rates for each frame type

$$R = R_I + R_P + R_B. \tag{11}$$

Specifically, when no frames are discarded due to temporal scaling, using the above equations for R_I, R_P, and R_B, the total playable frame rate, R, is:

$$\begin{aligned} R &= G \cdot q_I + G \cdot q_I . \frac{q_P - q_P^{N_P+1}}{1-q_P} \\ &\quad + N_{BP} \cdot G \cdot q_I \cdot q_B \cdot \left(\frac{q_P - q_P^{N_P+1}}{1-q_P} + q_I \cdot q_P^{N_P} \right) \\ &= G \cdot q_I \cdot \left(1 + \frac{q_P - q_P^{N_P+1}}{1-q_P} + N_{BP} \cdot q_B \cdot \left(\frac{q_P - q_P^{N_P+1}}{1-q_P} + q_I \cdot q_P^{N_P} \right) \right). \end{aligned} \tag{12}$$

3.4 Optimal Playable Frame Rate

For given values of p, (N_P, N_B), and (S_I, S_P, S_B), the total playable frame rate R varies with the temporal scaling and the amount of FEC as a function R((N_{PD}, N_{BD}), (S_{IF}, S_{PF}, S_{BF})). In addition, given t_{RTT} and s, the total bitrate is also constrained by the TCP-friendly rate T in Equation (1)

$$G \cdot ((S_I + S_{IF}) + N_{PD} \cdot (S_P + S_{PF}) + N_{BD} \cdot (S_B + S_{BF})) \leq T. \tag{13}$$

Our model can be used to optimize the playable frame rate, R, under the TCP-friendly rate constraint using following equation:

$$\begin{cases} \textit{Maximize:} \\ \quad R = R((N_{PD}, N_{BD}), (S_{IF}, S_{PF}, S_{BF})) \\ \textit{Subject to:} \\ \quad G \cdot ((S_I + S_{IF}) + N_{PD} \cdot (S_P + S_{PF}) + \\ \qquad N_{BD} \cdot (S_B + S_{BF})) \leq T \\ \quad 0 \leq N_{PD} \leq N_P, 0 \leq N_{BD} \leq N_B \\ \quad 0 \leq S_{IF} \leq S_I, 0 \leq S_{PF} \leq S_P, 0 \leq S_{BF} \leq S_B. \end{cases} \tag{14}$$

Unfortunately, finding a closed-form solution for the nonlinear function R is difficult due to the many saddle points. However, given that the optimization problem is expressed in terms of integer variables over a restricted domain, a complete search of the constrained discrete space is feasible. With fixed input values for (p, RTT, s), (N_P, N_B), and (S_I, S_P, S_B), the space of possible values for (N_{PD}, N_{BD}), and (S_{IF}, S_{PF}, S_{BF}) (subject to the temporal scaling constraints given in Section 2.4) can be quickly searched to determine the FEC and temporal scaling patterns that yield the maximum TCP-friendly playable frame rate. Preliminary investigations with nonoptimized code show that using our model to find the best adjusted FEC and temporal scaling pattern for GOP(3,8) takes about 30ms on a P-3 800 MHz. Note, this is much less than the real-time playout of 400ms for the GOP, it is even less than the playout time of a single frame, and it is even less than a typical feedback interval for network parameters which are normally updated every RTT.

4. ANALYTIC EXPERIMENTS

This section considers the design of a set of experiments that use the analytic model of playable frame rate to explore the performance of temporally-scaled MPEG video without FEC, with fixed FEC, and with adjusted FEC where the videos' bitrates are constrained by TCP-friendly data rates.

The MPEG video without FEC has the advantage of not adding overhead to the MPEG data packets and uses the full available bandwidth to transmit application data. However, this scheme is highly vulnerable to packet loss.

Table III. System Parameter Settings

Network Layer		MPEG Layer			
t_{RTT}	50 ms	S_I	24.6 Kbytes (25 pkts)	N_P	3 frames per GOP
s	1 Kbyte	S_P	7.25 Kbytes (8 pkts)	N_B	8 frames per GOP
p	0.01 to 0.04	S_B	2.45 Kbytes (3 pkts)	R_F	30 frames per sec

The fixed FEC strategy, denoted by FEC($S_{IF}/S_{PF}/S_{BF}$), uses a fixed amount of redundancy to protect the corresponding I, P, or B frames. This mechanism has the advantage of being resilient to specific packet loss but has the disadvantage of a reduced MPEG data rate due to the FEC overhead.

Using the equations in Section 3, the adjusted FEC algorithm selects the FEC and temporal scaling patterns that achieve the maximum playable frame rate. This technique has the advantage of providing the amount of FEC appropriate for the current network conditions, but it does not perform well outside of the analytic model and requires a more complex implementation. Section 5 presents experiments conducted to evaluate the effectiveness of the model under more realistic network conditions.

As for the rest of this section, we first present our experimental methodology in Section 4.1 and our system settings in Section 4.2, and then our analysis in Section 4.3.

4.1 Methodology

To evaluate the derived equations with various parameter values, we built programs to implement them analytically. Using the formulas in Section 3, we built a function, `frameRate()` to use Equation (14) to compute the playable frame rate with given network characteristics (p, t_{RTT}, s), MPEG properties (N_P, N_B), (S_I, S_P, S_B), temporal scaling pattern (N_{PD}, N_{BD}), and amounts of FEC (S_{IF}, S_{PF}, S_{BF}).

Another program was built such that given values of (p, t_{RTT}, s), (N_P, N_B) and (S_I, S_P, S_B) the program searches through all combinations of FEC (S_{IF}, S_{PF}, S_{BF}), and temporal scaling patterns (N_{PD}, N_{BD}). Initially, each combination of FEC and scaling are tested to determine if this combination satisfies the TCP-friendly rate constraint (Equation (13)). If this combination does not satisfy the constraint, the search program goes to the next iteration. If the constraint is satisfied, the `frameRate()` function is used to determine the playable frame rate for this FEC and scaling combination. After searching all the combinations of FEC and scaling patterns within the constrained search space, the program produces the maximum playable frame rate, the adjusted FEC (S_{IF}, S_{PF}, S_{BF}), and the temporal scaling (N_{PD}, N_{BD}) required to achieve this maximum rate.

In Section 4.3, these programs are employed to explore frame rate performance over a range of network and MPEG settings. For each set of network and MPEG parameters, we compare the playable frame rate of MPEG video without FEC, MPEG video with fixed FEC, and MPEG video with adjusted FEC.

4.2 System Settings

Table III presents the system parameter settings for the network and MPEG layers. The MPEG frame sizes were chosen using the mean I, P, B frame sizes measured in Krunz et al. [1995] and then rounding up the frame size to the nearest integer number of packets. Specifically, the I frame has 25 packets, the P frame has 8 packets, and the B frame has 3 packets. A commonly used MPEG GOP pattern, "`IBBPBBPBBPBB`", (GOP(3,8)) and a typical full-motion frame rate R_F of 30 frames per second (fps) were used. These settings yield a packet rate of 146 packets per second and a data rate of 1.168 Mbps for the MPEG video. The packet size s, round-trip time t_{RTT} and packet loss probability p were chosen based on the characteristics of many network connections [Paxson 1999; Chung et al. 2003; Jaiswal et al. 2004]. For all experiments, the parameters are fixed, except for the packet loss probability p which ranges from 0.01 to 0.04 in steps of 0.001.

4.3 Analysis

We analyze the playable frame rate for non-FEC, fixed FEC, and adjusted FEC MPEG video and explain the effects of FEC and temporal scaling.

4.3.1 *Playable Frame Rate.* We compare the playable frame rates for four distinct repair schemes.

(1) Fixed FEC (1/0/0). Each I frame receives 1 FEC packet. This simple FEC pattern protects the most important frame, the I frame. Repairing the I frame is a scheme used by other researchers [Feamster and Balakrishnan 2002; Rhee 1998].
(2) Fixed FEC (4/2/1). The sender protects each I frame with 4 FEC packets, each P frame with 2 FEC packets, and each B frame with 1 FEC packet. This FEC pattern provides strong protection to each frame and roughly represents the relative importance of the I, P, and B frames. For the MPEG settings in Table III, this adds approximately 15% overhead for each type of frame which is typical for many fixed FEC approaches [Hardman et al. 1995; Hartanto and Sirisena 1999; Liu and Claypool 2000].
(3) Adjusted FEC. Before transmitting, the sender uses the program described in Section 4.1 to determine the FEC and temporal scaling patterns that produce the maximum playable frame rate and uses these for the entire video transmission.
(4) Non-FEC. The sender adds no FEC to the video.

In all cases, the total bandwidth used by the MPEG video plus FEC is temporally scaled (as described in Section 2.4) to meet TCP-friendly constraints.

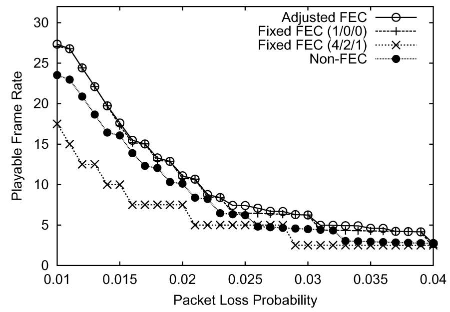
While there are numerous other fixed FEC and MPEG video choices that could be selected, due to space limitations, we only present the analysis of the four representative systems given. However, the fact that these choices include commonly used FEC patterns and the parameters were chosen to capture typical MPEG characteristics justifies this method of performance comparison. Moreover, while other fixed FEC patterns may do as well as adjusted FEC for some MPEG videos under a given set of network conditions, fixed FEC schemes cannot operate effectively over the full range of typical MPEG and network parameters. However, additional comparisons that include other fixed FEC schemes can be found in Wu et al. [2003b].

Figure 2 depicts the playable frame rates for each of the four schemes. For all figures, the x-axes are the packet loss probabilities, and the y-axes are the playable frame rates. For frame rate targets [Real Networks Incorporated 2000]. 24-30 frames per second is full-motion video, 15 frames per second can approximate full-motion video for some video content, 7 frames per second appears choppy, and at 3 frames per second or below, the video becomes a series of still pictures.

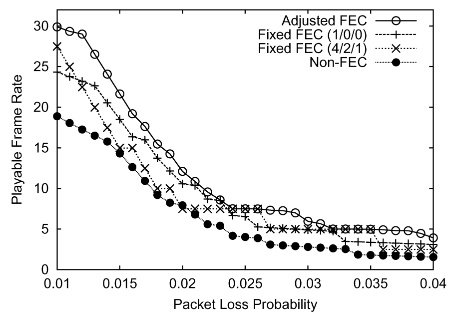
In Figure 2, adjusted FEC provides the highest playable frame rate under all network and video conditions. For the typical video size in Figure 2b, the benefits of adjusted FEC over non-FEC are substantial, almost doubling the frame rate at 1% loss and still surpassing the minimum 2 frames per second at 4% loss. The two fixed FEC techniques usually improve playable frame rates over non-FEC video, and FEC(4/2/1) even matches the playable frame rate provided by adjusted FEC for a few loss rates, such as 2.5%.

For smaller video frame sizes in Figure 2a, halving the frame sizes in Table III and doubling the round-trip time to provide an available bandwidth allows a visual comparison between graphs. FEC(1/0/0) does substantially better, coming closer to the maximum frame rate achieved by adjusted FEC. FEC(4/2/1) does worse with playable frames below the non-FEC scheme. In this case, it happens because the fixed number of FEC packets added is a larger fraction of overhead for the smaller video frames.

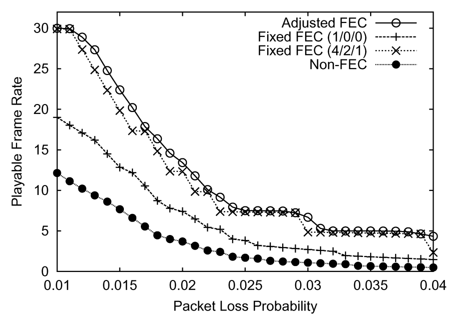
For the larger video frame sizes in Figure 2c created by doubling the frame sizes in Table III and halving the round-trip time, FEC(4/2/1) does substantially better and provides close to the maximum



a. Small Frame Size (1/2 those in Table III)



b. Medium Frame Size (as in Table III)



c. Large Frame Size (2x those in Table III)

Fig. 2. Comparison of playable frame rates.

frame rate achieved by adjusted FEC. FEC(1/0/0) does significantly worse since it does not provide enough protection for the larger frame sizes. With playable frame rates well below that of adjusted FEC, FEC(1/0/0) still outperforms the non-FEC scheme.

These figures show fixed FEC only works well for specific network and MPEG conditions. For example, FEC(1/0/0) works nearly as well as the adjusted FEC in Figure 2a, while FEC(4/2/1) works nearly as well as the adjusted FEC in Figure 2c. However, when the network and MPEG conditions change, both fixed FEC patterns chosen are less effective than the more robust adjusted FEC scheme. This general behavior holds for other fixed FEC choices regardless of the specific input patterns used.

4.3.2 *Adjusting FEC.* To better explain the benefits of adjusted FEC presented in the previous section, we now analyze how FEC is adjusted for various fixed loss rates.

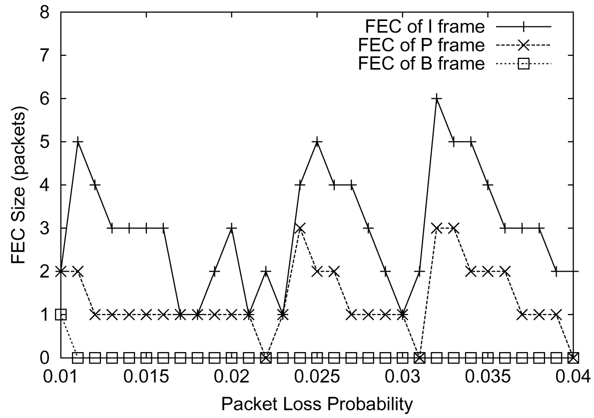


Fig. 3. Adjusted FEC pattern.

Table IV. Temporal Scaling Patterns

p	Adjusted FEC	Non- FEC
0.010	`IBBPBBPBBPBB`	`IBBPBBPBBPBB`
0.015	`IBBPB-PB-PB-`	`IBBPBBPBBPB-`
0.020	`IB-P--P--P--`	`IB-PB-PB-P--`
0.025	`I--P--P-----`	`I--P--P--P--`
0.030	`I--P--P-----`	`I--P--P-----`
0.035	`I--P--------`	`I--P--------`
0.040	`I--P--------`	`I--P--------`

Figure 3 gives the breakdown of the adjusted FEC for each I, P, and B frame that produces the maximum playable frame rate versus the loss probability. The fixed FEC approaches are not shown, but they would be represented by horizontal lines since they introduce the same amount of FEC for all loss probabilities. For example, FEC(4/2/1) would have a horizontal line at 4 for the I frames, at 2 for the P frames, and at 1 for the B frames. In general, without FEC, I frames have a decreasing probability of successful transmission. With adjusted FEC, the most important I frames have the highest transmission probability, followed by the P frames, and lastly, by the least important B frames. However, there are cases where the best use of FEC is somewhat nonintuitive. For instance, at 1.7% loss, the adjusted FEC scheme reduces the FEC for the Iframes and then increases it at 1.9%. This seeming contradiction is because the use of FEC is coupled with temporal scaling. In particular, at 1.7%, the playable frame rate is higher if four B frames are transmitted (transmitting "`IB-PB-PB-PB-`"), leaving less leftover capacity for FEC. At the increased loss rate of 1.9%, the reduced available bandwidth and higher loss rates makes discarding two more B frames (transmitting "`IB-PB-P--P--`") and using the remaining bandwidth for FEC the right choice for a higher playable frame rate.

4.3.3 *Temporal Scaling Pattern.* Table IV shows the chosen temporal scaling pattern for adjusted FEC as loss probability varies. The "-" symbol denotes frames that are discarded by the sender before being transmitted. A B frame is automatically discarded if the following P frame it references is discarded. Although there may be available capacity for the transmission, this B frame still cannot be displayed by the receiver and thus it is discarded. As p increases, the available bitrate under the TCP-friendly constraint decreases, and the sender discards the less important frames before sending them. The I frames are always transmitted, the P frames are kept as long as possible, and the B frames are discarded before the P frames they reference. In general, MPEG video with adjusted FEC must discard slightly more frames than the same MPEG video without FEC. However, the additional packet space saved by the discards can be very effectively used for FEC packets. Temporal scaling patterns over a larger range of packet loss probability can be found in Wu et al. [2003b].

Note that the temporal scaling patterns in Table IV may result in a variable playable frame rate when measured over one GOP. Our future work is to incorporate the impact of variance in frame rates into our model and get the optimal scaling pattern for the best perceived quality. If a low variance is more important than a high playable frame rate, only scaling patterns that evenly distribute the frame discards can be considered.

5. SIMULATION EXPERIMENTS

Our model is intended for use as the core of a streaming protocol that adjusts FEC and temporal scaling in response to real-world application and network conditions. For the experiments in Section 4, the MPEG layer and network layer parameters remained fixed for the duration of each video. This simplified environment allowed us to clearly illustrate the effects of adjusted FEC compared to that of fixed FEC and non-FEC approaches. However, in practice, MPEG video frame sizes change over the course of a video, and they may even change in the middle of a GOP. Moreover, while maximum network packet sizes are often fixed for the life of a flow, round-trip times and loss rates change rapidly, and packet losses are often bursty.

This section explores our model's accuracy in predicting playable frame rate by designing simulation experiments that characterize more realistic network and video conditions. Comparing performance predicted by the model against simulated performance provides a strong indication of the effectiveness of using our model within a streaming protocol in real Internet situations. Specifically, the analytic experiments assumed:

(1) an accurate estimate of the packet loss probability from the network protocol. Section 5.1 considers the effects of error in the packet loss estimate on our model's predictive quality.
(2) independent network packet losses. Section 5.2 introduces bursty packet losses derived from previous Internet streaming measurements to determine the impact of the independent packet loss assumption on our model's accuracy.
(3) fixed round-trip times for the life of the flow. Section 5.3 uses our model to determine the appropriate temporal scaling assuming fixed round-trip times and then applies more realistic round-trip times obtained from traces of Internet streaming experiments.
(4) constant I, P, and B frame sizes for the entire video. Section 5.4 uses our model assuming a fixed frame size and then applies more realistic frame sizes based on traces from previous measurements of MPEG video.

For each experiment, the playable frame rate predicted by our analytic model is compared to the actual frame rate achieved through the more realistic simulations. The comparison of the estimated playable frame rate to the actual frame rate achieved shows how sensitive our model is to real-world effects, while comparisons of the playable frame rate with fixed FEC or without FEC indicate the advantages of using our model even if there are real-world inaccuracies. For all experiments, the system parameters that are not varied are the same as in Table III. For example, the round-trip time used is 50ms. Depending on the input loss rate used, this yields a TCP-friendly bandwidth ranging from 0.71 Mbps to 1.80 Mbps.

5.1 Inaccurate Loss Prediction

This simulation tests the effectiveness of using the adjusted FEC determined by the model when the loss rate is not accurately predicted. While underpredicting the loss rate results in too little FEC for

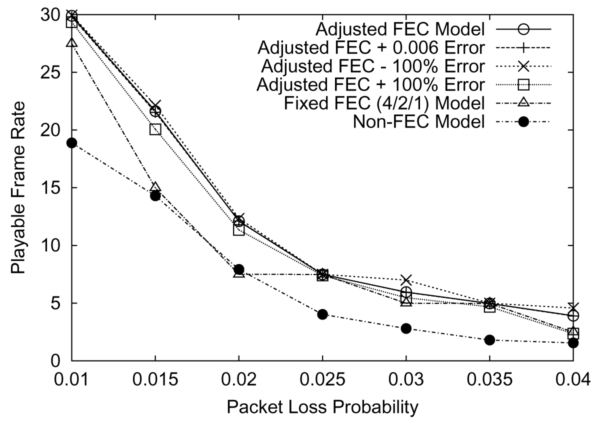


Fig. 4. Impact of inaccurate loss prediction.

effective repair, overpredicting the loss rate yields more FEC than necessary and leaves less available bitrate for the MPEG data. Three sets of simulation experiments with different induced amounts of error in the loss probability prediction were run: 1) the actual loss rate was higher than the predicted loss rate by 0.6% which is the average margin for error found after numerous simulations in Floyd et al. [2000]; 2) the actual loss rate was double the predicted loss rate; and 3) the actual loss rate was half the predicted loss rate.

For each loss case, the predicted loss rate p was used in the adjusted FEC model to determine the FEC and temporal scaling patterns. Then, we simulated streaming the MPEG video using these patterns on a network with the actual losses and measured the actual playable frame rate at the receiver.

Figure 4 depicts the playable frame rates for the simulations along with the playable frame rates estimated by our model. For the cases in which the actual error was underestimated, our model's frame rate estimate does differ from the actual frame rate achieved, indicating that the inaccurate loss prediction does result in a slightly suboptimal use of FEC. However, the actual frame rates achieved differ by less than 0.5 frame per second on average. Moreover, for the practical loss prediction errors of 0.006, the actual frame rates are nearly identical to the predicted frame rates. This suggests using our model to determine proper FEC and temporal scaling can be effective in practice.

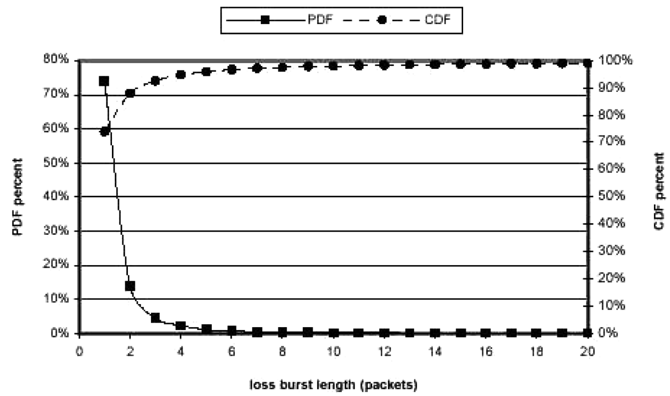
5.2 Bursty Loss

Our analytic model assumes independent packet loss events, while Internet packet losses are often bursty [Loguinov and Radha 2001; Paxson 1999]. Bursty losses may reduce the effectiveness of FEC especially when fewer than K of the N packets in a frame can be recovered and the resultant playable frame rate is lowered.

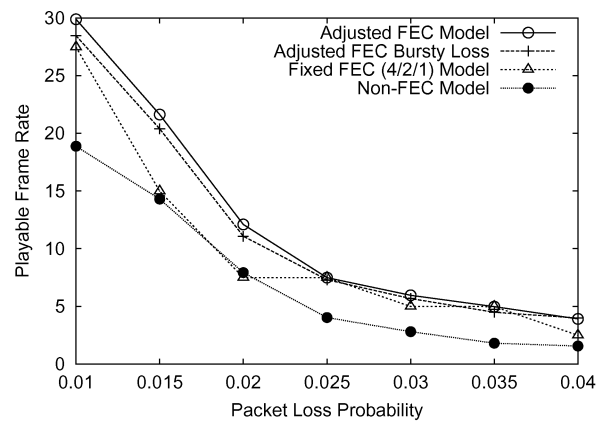
We used a series of traces from an Internet measurement study [Chung et al. 2003] to simulate the effects of bursty loss over a range of loss conditions. For each loss event, we used the probability distribution obtained from Internet streaming traces in Loguinov and Radha [2001] and depicted in Figure 5a to provide bursty loss events.

We used our model to determine the adjusted FEC and predicted frame rate assuming independent losses. Then, we simulated streaming the MPEG video using the trace-driven loss events and loss bursts and measured the actual playable frame rate at the receiver.

Figure 5 depicts the playable frame rates for the simulations along with the playable frame rates estimated by our model. The bursty packet loss simulations do show that the adjusted FEC model with



a. Loss Burst Distribution (from [Loguinov and Radha 2001])



b. Playable Frame Rate

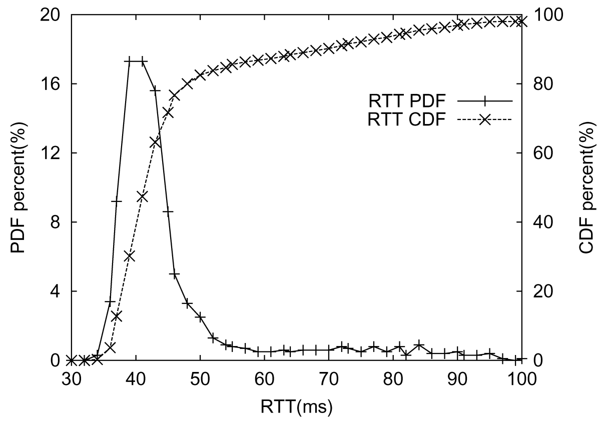
Fig. 5. Impact of bursty loss.

independent loss assumptions predicts marginally overoptimistic performance. However the differences are small enough to suggest that using the model to determine adjusted FEC, based on independent losses, yields good performance in practice.

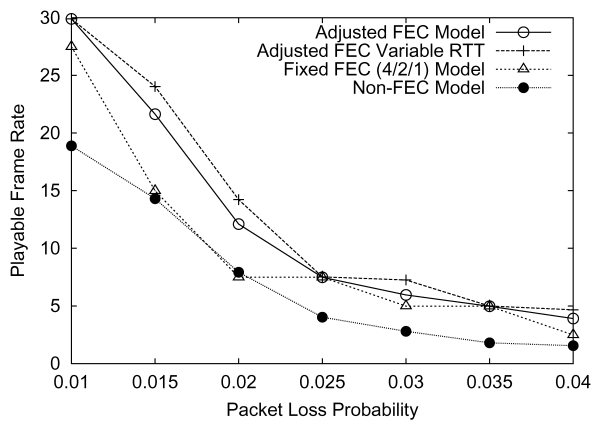
5.3 Variable Round-Trip Times

Our analytical model assumes fixed round-trip times (RTTs) for the entire flow. In reality, RTTs can vary considerably. The possible impact of variable RTTs is that the bandwidth estimate using a fixed average RTT is inaccurate, and therefore this causes the choices for temporal scaling and FEC to be less effective.

To study the effects of variable RTTs, we selected a trace from Chung et al. [2003], depicted in Figure 6a, that had an average RTT of about 45 milliseconds. We used our model to determine the adjusted FEC and temporal scaling patterns assuming a fixed RTT of 50 milliseconds. Then, we simulated streaming the MPEG video using the RTT trace and measured the actual frame playout rate at the receiver. To make the results comparable, each RTT from the trace is multiplied by 50/45 before the simulation so the average RTT of the simulation becomes 50 milliseconds.



a. RTT Distribution (from [Chung et al. 2003])



b. Playable Frame Rate

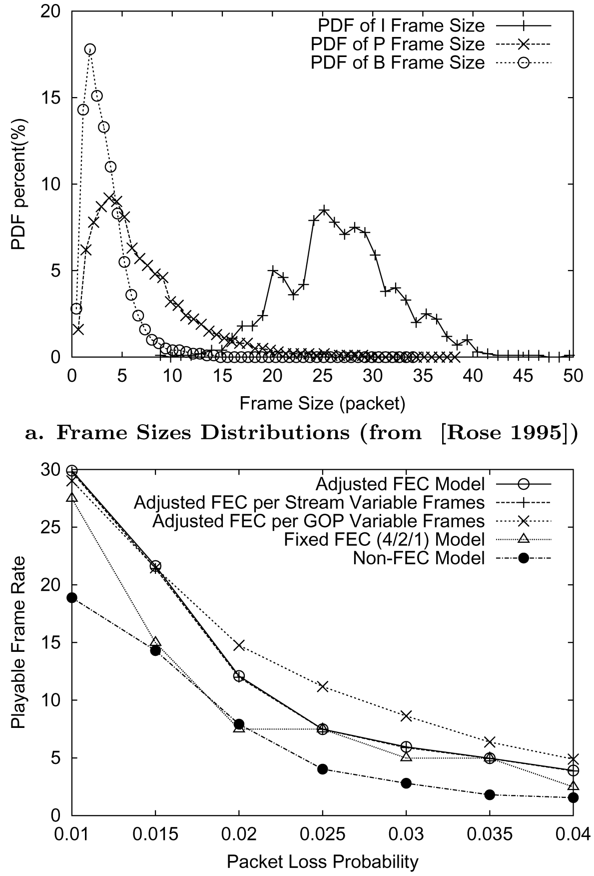
Fig. 6. Impact of variable RTT.

Figure 6b depicts the playable frame rates for the simulations along with the playable frame rates estimated by our model. Surprisingly, the variable RTT curve has a slightly higher playable frame rate than our model estimated by using the average RTT. We attribute this to the fact that the RTT distribution selected is not Gaussian (normal) but instead has a somewhat heavy tail. Overall, even though the RTTs cover a wide range, the playable frame rate estimated by our model is close to the actual playable frame rate, further suggesting that our model can be effective in practice.

5.4 Variable MPEG Frame Sizes

In the development of the analytic model, the MPEG frame size is assumed constant for the entire video. In reality, MPEG frame sizes change constantly, and they may even change inside one GOP. There are two possible impacts of variable-sized frames on the accuracy of the model: (1) the adjusted FEC chosen using fixed average frame sizes will be inappropriate for the actual frame sizes and result in a lower playable frame rate; (2) our model will have to be applied separately for each GOP to chose the appropriate FEC adjustment. This adds increased overhead to the streaming application.

To simulate the effects of variable MPEG frame sizes, we selected a frame size trace from Rose [1995]. Figure 7a presents the PDF distributions for frame types from this trace.



a. Frame Sizes Distributions (from [Rose 1995])

b. Playable Frame Rate

Fig. 7. Impact of variable frame size.

Once again, the model was used to determine the adjusted FEC and temporal scaling pattern assuming a fixed average frame size. Then, we simulated streaming the MPEG video using the frame size trace and determined the actual playable frame rate at the receiver. Additionally, we applied the model to each individual GOP, thus computing a new adjusted FEC based on the current GOP's I, P, and B frame sizes. We simulated streaming the MPEG video using this per GOP adjusted FEC and measured the playable frame rate at the receiver.

Figure 7b graphs the playable frame rates for the simulations along with the playable frame rates estimated by our model. The frame rate depicted by the adjusted FEC model is almost the same as the adjusted FEC per stream simulation. At 2.0% loss rate and above, the simulation of adjusted FEC per GOP simulation produces a higher playable frame rate than all of the curves in Figure 7b.

Figure 8 focuses on the specific case of 2.5% loss to compare the simulated FEC scheme against data rates produced by the model. Since the model uses a fixed frame size, it yields a constant data rate equal to the TCP-friendly rate of 126 packets per second. While remaining TCP-friendly over long time periods, the adjusted FEC per stream simulation produces considerable variation in its data rate. The adjusted FEC per GOP simulation, however, has a much smoother data rate that is significantly closer

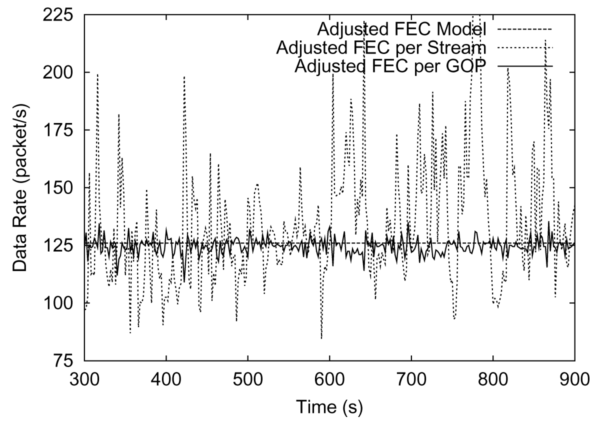


Fig. 8. Data rates for 2.5% loss rate.

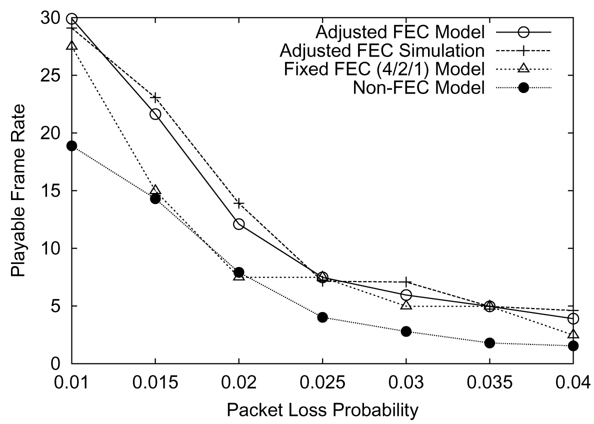


Fig. 9. Combination effects.

to the predicted constant data rate. Note, smooth data rates are much easier for networks to manage than bursty data rates.

Based on the observation from Figure 7 and Figure 8 where adjusted FEC per GOP simulation has higher playable frame rate and smoother data rate, we suggest that the adjusted FEC model should be applied to every GOP. Since one instance of the model calculation can be executed in much less time than the real-time playout time for a GOP (Section 3.4), this repetitive use of the model is feasible.

5.5 Combination Effects

The previous sections show the model to be resilient to mispredictions for loss rate, RTT, or frame sizes. In this section, we simulated the effect of the combination of all three of these mispredictions on the model. Specifically, simulation loss rate is set higher than the predicted loss rate by 0.6%, the simulation RTT is from the RTT trace in Section 5.3, and the simulation actual sizes come from the frame size trace in Section 5.4. The results from this simulation are compared to the analytic model performance where the model assumes a fixed loss rate, a fixed RTT, and fixed frame sizes.

Figure 9 presents the playable frame rates for the simulations along with the playable frame rates estimated by our model. The combined effect is very similar to the RTT effect seen in Figure 6b where

Table V. Misprediction Errors

	Loss Rate						
	0.010	0.015	0.020	0.025	0.030	0.035	0.040
Loss rate + 0.006 error	−0.09	−0.10	0.02	0.00	−0.02	−0.01	−0.02
Loss rate − 100% error	−0.54	−1.57	−0.76	−0.09	−0.49	−0.30	−1.56
Loss rate + 100% error	0.06	0.50	0.24	0.00	1.04	0.02	0.67
Bursty Loss	−1.42	−1.25	−1.02	−0.18	−0.29	−0.49	0.07
Variable RTT	0.00	2.40	2.12	0.04	1.29	0.02	0.76
Variable Frame per Stream	−0.11	−0.21	−0.11	−0.01	−0.07	−0.02	−0.03
Variable Frame per GOP	−0.88	−0.16	2.66	3.69	2.67	1.40	0.98
Combination Effects	−0.81	1.43	1.80	−0.33	1.13	0.00	0.71

the simulation has a slightly higher playable frame. The obvious conjecture is that the RTT effect dominates the combination effect. Overall, the playable frame rate estimated by our model is close to the simulated playable frame rate. This provides further evidence that the model can be used in practice.

5.6 Practical Considerations

Table V provides a summary of the misprediction errors presented in the previous sections. The first three lines show the errors from loss rate mispredictions from experiments discussed in Section 5.1. The fourth line shows the errors from bursty loss considered in Section 5.2. The fifth line shows the errors from the variable RTT experiments in Section 5.3. The next two lines show the errors from variable frame sizes, and the last line shows the errors from a combination of mispredictions.

If our model is to be used for interactive video, there are three practical issues that need to be addressed: (1) dynamically changing a GOP based on the network conditions; (2) arbitrarily long GOPs that would make exhaustive search prohibitive; and (3) the case when the network and MPEG parameters are not known ahead of time. We address each concern separately:

(1) For interactive streaming media encoded on the fly, the encoder can change the GOP (e.g., vary the number of P and B frames in the GOP). However, our prior investigation [Wu et al. 2003a] demonstrates that with FEC and a typical GOP (such as IBBPBBPBBPBBPBB), adjusting the GOP does not improve the playable frame rate.
(2) In theory, GOP's can be arbitrarily long. However, our recent research [Wu et al. 2004] explains how, in practice, the GOP length can be effectively bounded. These previous results consistently suggest two guidelines: (a) the number of B frames between two reference frames should be only one or two (except when limited further by the encoding and time constraints); and (b) there is little performance gain in having more than five P frames per GOP.
(3) If the network and MPEG parameters are known in advance, a system can use our model to precompute the optimal FEC and temporal scaling pattern for some typical network conditions. If the network and MPEG parameters are not known in advance (such as for an interactive videoconference), the streaming application can keep weighted moving average estimators of the MPEG frame sizes, the packet loss rates, and flow round-trip times from the previous epoch as an estimate of the parameters to use for the next epoch. While these estimators are likely to introduce mispredictions, all the experiments in this section indicate that using the analytic model with estimated (and therefore somewhat inaccurate) parameters still yields optimal playable frame rates that are within 1.8 frame per second of model estimates where all parameters are known in advance. Moreover, the concept of locality of frame sizes in relation to adjacent GOP's in an epoch

[Garret and Willinger 1994] would lead one to believe that an estimator based on a weighted moving average would be reasonably accurate.

6. CONCLUSIONS

This article proposes an analytic model for a TCP-friendly MPEG stream that captures the dependencies between MPEG frame types and computes the playable frame rate of temporally-scaled MPEG video with Forward Error Correction (FEC) in the presence of packet loss. We use this model to determine the optimal adjustment of FEC and temporal scaling taking into account both current network conditions and application settings.

The analytic experiments presented indicate that adjusting FEC with temporal scaling provides improvement over current approaches. The Adjusted FEC mechanism always achieves a higher playable frame rate than MPEG video without FEC and provides a higher playable frame rate than any fixed FEC approaches when taken over a wide range of possible MPEG encoding and network conditions. Our simulation experiments show that using the model to maximize playable frame rate is reasonable over a range of realistic system conditions including inaccurate loss predictions, bursty packet losses, variable round-trip times, and variable MPEG frame sizes. The experimental results illustrate the feasibility of our model as the core of a streaming protocol layer that adapts the FEC and temporal scaling to the current system on the fly to provide substantial increases in playable frame rates, while maintaining a TCP-friendly bandwidth.

Ongoing work includes implementation of a streaming MPEG system that includes our model. This will allow us to perform Internet experiments and conduct users studies as well as carefully investigate the real-time properties of the system. Another area of ongoing research includes studying other measures of video performance, such as the MPQM model [van den Branden Lambrecht and Verscheure 1996] and the ITS-VQM model [Pinson and Wolf 2004]. This will allow enhancement of our model to include *quality scaling* such as in MPEG FGS (fine granularity scaling), where more lower quality frames are transmitted instead of sending fewer, higher quality frames. Previous work demonstrates that the choice of temporal or quality scaling can, in fact, account for the content of the video being streamed [Tripathi and Claypool 2002]. Other potential areas of future work include extending our model to analyze other types of media repair, such as media-dependent FEC as in Liu and Claypool [2000] and selective retransmissions as in Feamster and Balakrishnan [2002].

REFERENCES

Acharya, S. and Smith, B. 1998. An experiment to characterize videos stored on the Web. In *Proceedings of the ACM/SPIE Multimedia Computing and Networking (MMCN)*. San Jose, CA, 166–178.

Albanese, A., Bomer, J., Edmonds, J., Luby, M., and Sudan, M. 1996. Priority encoding transmission. *IEEE Trans. Inform. Theory 42*, 6 (Nov.), 1737–1744.

Balakrishnan, H., Rahul, H., and Seshan, S. 1999. An integrated congestion management architecture for Internet hosts. In *Proceedings of ACM SIGCOMM*. Cambridge, MA.

Bocheck, P., Campbell, A., Chang, S.-F., and Lio, R. 1999. Utility-based network adaptation for MPEG-4 systems. In *Proceedings of International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*. Basking Ridge, NJ.

Bolot, J.-C., Fosse-Parisis, S., and Towsley, D. 1999. Adaptive FEC-based error control for Internet telephony. In *Proceedings of IEEE INFOCOM*. New York, NY.

Boutremans, C., Iannaccone, G., and Diot, C. 2002. Impact of link failures on VoIP performance. In *Proceedings of the Network and Operating System Support for Digital Audio and Video (NOSSDAV)*. Miami Beach, FL.

Boyce, J. and Gaglianello, R. 1998. Packet loss effects on MPEG video sent over the public Internet. In *Proceedings of ACM Multimedia*. Bristol, U.K., 181–190.

Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., L. Peterson, K. R., Shenker, S., Wroclawski, J., and Zhang, L. 1998. Recommendations on queue management and congestion avoidance in the Internet. *IETF Request for Comments (RFC)* 2309.

Chung, J., Claypool, M., and Zhu, Y. 2003. Measurement of the congestion responsiveness of RealPlayer streaming video over UDP. In *Proceedings of the Packet Video Workshop (PV)*. Nantes, France.

DeSantis, T. and Loose, D. 2003. TCP traffic analysis. Major Qualifying Project MQP-MLC-MT03 Worcester Polytechnic Institute. Advisors M. Claypool and R. Kinicki.

Feamster, N. and Balakrishnan, H. 2002. Packet loss recovery for streaming video. In *Proceedings of 12th International Packet Video (PV) Workshop*. Pittsburgh, PA.

Floyd, S. 1994. TCP and explicit congestion notification. *Comput. Comm. Rev.*

Floyd, S. and Fall, K. 1999. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Trans. Netw.*

Floyd, S., Handley, M., Padhye, J., and Widmer, J. 2000. Equation-based congestion control for unicast applications. In *Proceedings of ACM SIGCOMM Conference*. Stockholm, Sweden, 43–56.

Garret, M. W. and Willinger, W. 1994. Analysis, modeling and generation of self-similar VBR video traffic. In *Proceedings of ACM SIGCOMM*.

Hardman, V., Sasse, M. A., Handley, M., and Watson, A. 1995. Reliable audio for use over the Internet. In *Proceedings of Internet Society's International Networking Conference (INET)*.

Hartanto, F. and Sirisena, H. R. 1999. Hybrid error control mechanism for video transmission in the wireless IP networks. In *Proceedings of the 10th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN)*.

Hemy, M., Hangartner, U., Steenkiste, P., and Gross, T. 1999. MPEG system streams in best-effort networks. In *Proceedings of Packet Video Workshop*.

International Telecommunications Union. 1996. One-way transmission time. Tech. Rep. G.114, ITU-T Recommendation.

Jaiswal, S., Iannaccone, G., Diot, C., Kurose, J., and Towsley, D. 2004. Inferring TCP connection characteristics through passive measurements. In *Proceedings of IEEE Infocom*. Hong Kong, China.

Krunz, M., Sass, R., and Hughes, H. 1995. Statistical characteristics and multiplexing of MPEG streams. In *Proceedings of IEEE Infocom*. Boston, MA, 455–462.

Liu, Y. and Claypool, M. 2000. Using redundancy to repair video damaged by network data loss. In *Proceedings of IS&T/SPIE/ACM Multimedia Computing and Networking (MMCN)*.

Loguinov, D. and Radha, H. 2001. Measurement study of low-bitrate internet video streaming. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*.

Mahajan, R., Floyd, S., and Wetherall, D. 2001. Controlling high-bandwidth flows at the congested router. In *Proceedings of the 9th International Conference on Network Protocols (ICNP)*.

Mayer-Patel, K., Le, L., and Carle, G. 2002. An MPEG performance model and its application to adaptive forward error correction. In *Proceedings of ACM Multimedia*.

Mena, A. and Heidemann, J. 2000. An empirical study of real audio traffic. In *Proceedings of IEEE Infocom*. Tel-Aviv, Israel, 101–110.

Mitchell, J. and Pennebaker, W. 1996. *MPEG Video: Compression Standard*. Chapman and Hall.

Nguyen, T. and Zakhor, A. 2002. Distributed video streaming with forward error correction. In *12th International Packet Video Workshop*.

Padhye, C., Christensen, K., and Moreno, W. 2000. A new adaptive FEC loss control algorithm for voice over IP applications. In *Proceedings of IEEE International Performance, Computing and Communication Conference*. Phoenix, AZ.

Padhye, J., Firoiu, V., Towsley, D., and Kurose, J. 1998. Modeling TCP throughput: A simple model and its empirical validation. In *Proceedings of ACM SIGCOMM*. Vancouver, Brisish Columbia, Canada.

Park, K. and Wang, W. 1999. QoS-sensitive transport of real-time MPEG video using adaptive forward error correction. In *Proceedings of IEEE Multimedia Systems*. 426–432.

Paxson, V. 1999. End-to-end internet packet dynamics. *IEEE/ACM Trans. Networ.*

Pinson, M. and Wolf, S. 2004. A new standardized method for objectively measuring video quality. *IEEE Trans. Broadcast. 50*, 3 (Sept.), 312–322.

Real Networks Incorporated. © 2000. RealProducer User's Guide. URL: http://www.service.real.com/help/library/guides /producerplus85/producer.htm.

Reed, I. S. and Solomon, G. 1960. Polynomial codes over certain finite fields. *J. SIAM Society Indust. Appl. Math. 8*, 2 (June), 300–304.

Rejaie, R., Handley, M., and Estrin, D. 1999. RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet. In *Proceedings of IEEE Infocom*. New York, NY, 1337–1345.

Rhee, I. 1998. Error control techniques for interactive low-bit rate video transmission over the internet. In *Proceedings of ACM SIGCOMM Conference*.

Rizzo, L. 1997. On the feasibility of software FEC. Tech. Rep. LR-970131, DEIT.

Rose, O. 1995. Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems. In *Proceedings of 20th Conference of Local Computer Networks*. 397–406.

Tripathi, A. and Claypool, M. 2002. Improving multimedia streaming with content-aware video scaling. In *Workshop on Intelligent Multimedia Computing and Networking (IMMCN)*. Durham, NC.

van den Branden Lambrecht, C. J. and Verscheure, O. 1996. Quality measure using a spatio-temporal model of the human visual system. In *Proceedings of SPIE*.

Wang, Y., Claypool, M., and Zuo, Z. 2001. An empirical study of RealVideo performance across the Internet. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop (IMW)*. San Francisco, CA. 295–309.

Wu, H., Claypool, M., and Kinicki, R. 2003a. A model for MPEG with forward error correction and TCP-friendly bandwidth. In *Proceedings of Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*. Monterey, CA.

Wu, H., Claypool, M., and Kinicki, R. 2003b. Adjusting forward error correction with temporal scaling for TCP-friendly streaming MPEG. Tech. Rep. WPI-CS-TR-03-10, CS Department, Worcester Polytechnic Institute.

Wu, H., Claypool, M., and Kinicki, R. 2004. Practical group of pictures considerations. In *Proceedings of the Packet Video Workshop (PV)*. Irvine, CA. Poster.

Received February 2005; revised June 2005; accepted August 2005