4-2003

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

Huahui Wu
Worcester Polytechnic Institute, flashine@cs.wpi.edu

Mark Claypool
Worcester Polytechnic Institute, claypool@wpi.edu

Robert Kinicki
Worcester Polytechnic Institute, rek@wpi.edu

WPI-CS-TR-03-10 April 2003

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

by

Huahui Wu
Mark Claypool
Robert Kinicki

Computer Science
Technical Report
Series

WORCESTER POLYTECHNIC INSTITUTE

Computer Science Department
100 Institute Road, Worcester, Massachusetts 01609-2280

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

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

April 14, 2003

Abstract

New TCP-Friendly requirements expect 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 add repair frames damaged by packet loss. However, adding FEC requires further reduction of the 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 paper, we derive analytically model the playable frame rate of a TCP-Friendly MPEG stream with FEC and temporal scaling. Our model captures the impact of specific FEC amounts for different MPEG frame types and accounts for 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 practically effective as part of a streaming protocol that chooses FEC and temporal scaling patterns that meet the current application and network conditions.

1 Introduction

As the number of active Internet users continues to grow and streaming media applications become more commonplace, the number of users and volume of data traversing the Internet is increasing in explosive proportions. The sheer number of possible number of users and applications at any point in time raises the probability of streaming multimedia flows encountering congestion. 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 dominant transport protocol on the Internet.

While streaming flows have traditionally selected UDP over TCP [23, 37], there is a growing consensus that all Internet applications must be TCP-Friendly,[1] and there are proposed approaches to detect and restrict non-TCP friendly flows [22, 11]. Thus, networking researchers have proposed new TCP-Friendly protocols (e.g. TFRC) [4, 13, 31, 33] for transporting streaming multimedia flows. By requiring TCP-Friendly streaming protocols, the belief is that router Active Queue Management techniques can be more effective to all forms of congestion. This, in turn, should yield better overall

[1] 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.

quality of service for streaming flows.

To preserve real-time streaming media playout, multimedia servers must scale back the data rate of their streaming flows to match perceived network conditions. This proactive data rate reduction by the multimedia server is called *media scaling* [5, 36]. 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 will only randomly drop packets [17]. *Temporal scaling* is a widely used form of media scaling whereby the multimedia server selectively discards frames prior to transmission.

While video applications can tolerate some data loss, excessive packet loss during congestion yields unacceptable media quality. Since multimedia encoding involves interframe dependencies [24], the random dropping of packets by routers can seriously degrade multimedia 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 can result in a 3%packet loss rate causing a 30% frame loss rate [7].

While TCP reacts to packet losses with retransmissions, applications such as videoconferencing and interactive virtual reality cannot afford the increased latency required for retransmissions, especially for high round trip times (RTTs). This suggests utilizing lower latency approaches, such as Forward Error Correction (FEC), in conjunction with TCP-Friendly protocols to deliver streaming applications over the Internet. Used properly, FEC [6, 26, 28, 35] can reduce or eliminate packet loss and partially or fully insulate video applications from degraded quality [19]. 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 addition of FEC data will reduce the effective transmission rate of 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 multimedia frames with inherent interframe encoding dependencies can be cast as a constrained optimization problem that attempts to optimize the quality of the video stream. Current approaches use either apriori, static FEC choices[15, 2] or adapt FEC to perceived packet loss on the network without regard to TCP-Friendly data rate constraints [28, 6, 26].

In this paper, we derive an analytic model that characterizes the performance of temporally scaled MPEG video with Forward Error Correction in the presence of packet loss. Given parameters to represent network loss, and MPEG frame types and sizes, our model allows specification of the number of FEC packets per MPEG frame type and temporal scaling patterns and computes the total playable frame rate. We represent a presentation layer network protocol by using 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 upon accurate estimates of packet loss and round-trip time, and upon fixed MPEG frame sizes, we design simulation experiments that study the effectiveness of using our model under realistic Internet conditions. The experimental results demonstrate that even with significant error in the estimated packet loss probability and with bursty packet losses, using our model to adjust FEC and temporal scaling patterns yields significant improvement in playable frame rate over current approaches. Also, since the analytic model assumes constant round-trip time and fixed MPEG frame sizes, we design additional simulation experiments with trace-driven round-trip times and MPEG frame sizes. The experimental results show that our model does a good job of selecting the FEC distribution for the video stream despite using only average round-trip times and fixed MPEG frame sizes. The cumulative effect of these experimental is to lend credibility to the fact that using our model to adjust FEC with temporal scaling can be effectively used to provide high playable frame rates for TCP-friendly streaming video.

The remainder of the paper is organized as follows: Section 2 provides background knowledge and clarifies terminology needed for the sections that follow; 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 and Section 7 summarizes the paper 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 in preparation for 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 (e.g., packet drop rate and round trip times). Padhye et al [27] analytically derived the following equation for TCP throughput:

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

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 sieze 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 [8, 12]. Thus, for the Internet to support future demands for multimedia applications, this research assumes transport protocols such as [13, 31, 33] that can keep multimedia streaming flows TCP-Friendly.

[2]Set to be $4t_{RTT}$ by [13]

2.2 Forward Error Correction (FEC)

Streaming video frames are often larger than a single Internet packet. Thus we model an application level video frame as being transmitted as K packets where K varies with frame type, encoding method, and media content. Media independent FEC [18] then 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.

To analyze the effects of FEC on application layer 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:

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

While this model ignores the bursty nature of Internet packet losses, we discuss the impact of this simplifying assumption in Section 5.

2.3 MPEG

The MPEG (Motion Picture Expert Group)[3] standard is gaining in popularity and appears a viable open standard for video on the Internet [24]. MPEG uses both intra-frame and inter-frame compression. I (intra-coded) frames are encoded independently and focus on encoding similarities within a video scene. P (predictive-coded) frames are encoded based on motion differences from preceding I or P frame in the video sequence. B (bi-directionally predictive-coded) frames are encoded based on motion differences from preceding and succeeding I or P frames.

MPEG video typically repeats a pattern of I, P, 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 in the figure marks the beginning of the next GOP and the arrows indicate frame dependency relationships. Because of

[3]http://mpeg.telecomitalialab.com/

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.

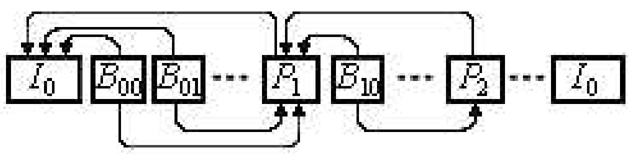


Figure 1: A sample MPEG Group Of Pictures

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[4]. 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'. In the rest of this paper, unless specifically indicated otherwise, GOP(3,8), a commonly used pattern on the Internet[1], will be used as the fixed GOP pattern. Analysis of other GOP patterns with similar results can be found in [38].

The subscripting scheme presented in Figure 1, will be used to identify individual frames within a GOP. The single I frame of a GOP is referred to as I_0, while P frames are named with P_i, where $1 \leq i \leq N_P$, and B frames are expressed as 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 bandwidth (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--`'.

A D subscript will be used to indicate the number of frames delivered after temporal scaling . Hence, N_{PD} denotes the number of P frames sent in one GOP, and N_{BD} denote the number of B frames delivered in one GOP (with $N_P - N_{PD}$ P frames discarded and $N_B - N_{BD}$ B frames 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.

While temporal scaling could select any of the frames in a GOP to discard, the following set of strategies that take into account MPEG frame dependencies will be used to minimize the ill effects of temporal scaling on the quality of the received video:

1. Since every frame in a GOP depends upon the I frame directly or indirectly, the I frame is the last frame discarded in a GOP.
2. Since each P frame depends upon the previous P frame or I frame, P frames are discarded from back to front position in the GOP pattern.
3. Since each B frame depends on the reference frames before and after it, B frames are all also discarded back to front.

Using these rules, for any GOP, $GOP(N_P, N_B)$, the temporal scaling reduction pattern can be uniquely identified by $M(N_{PD}, N_{BD})$. For example, temporal scaling $M(3,5)$ for $GOP(3,8)$ indicates the media server will send '`IBBPBBPB-P--`'.

To clarify the temporal scaling decision to deliver a specific frame in the formulation of the analytic model, we introduce a binary coefficient $D_{\#}$ (e.g. 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, implies frame # will be sent. Table 1 is presented as an example of the specific values of the D coefficients for $GOP(3,8)$ and $M(3,5)$ (the temporal scaling pattern '`IBBPBBPB-P--`').

3 Analytical Model

This section develops the analytic model used to determine the playable frame rate of TCP-Friendly

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

Table 1: A sample set of D values

#	I_0	B_{00}	B_{01}	P_1	B_{10}	B_{11}
$D_{\#}$	1	1	1	1	1	1
Pattern	I	B	B	P	B	B
#	P_2	B_{20}	B_{21}	P_3	B_{30}	B_{31}
$D_{\#}$	1	1	0	1	0	0
Pattern	P	B	-	P	-	-

streaming video flows with adjusted FEC and temporal scaling in the presence of network packet loss. First, we identify application, presentation and network parameters related to TCP-Friendly MPEG flows (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 a 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 System Layers and Parameters

Table 2: System Layers and Parameters

System Layer	Parameters
Application	$S_I, S_P, S_B, N_P, N_B, R_F$
Presentation	$S_{IF}, S_{PF}, S_{BF}, N_{PD}, N_{BD}$
Network	p, t_{RTT}, S

In our model, we incorporate the system layers and parameters indicated in Table 2, where the parameters are:

- R_F: the maximum playable frame rate achieved when there is enough bandwidth and no loss (typical full-motion video rates have $R_F = 30fps$).
- S_I, S_P, S_B: the size of I, P or B frames respectively in fixed size packets.
- N_P, N_B: the number of P, B frames in one GOP, respectively.
- N_{PD}, N_{BD}: the number of P and B frames (respectively) sent per GOP after temporal scaling.
- S_{IF}, S_{PF}, S_{BF}: the number of FEC packets added to I, P and B frame, respectively.
- S: the packet size (in bytes).
- p: the packet loss probability.
- t_{RTT}: the round-trip time.

For a streaming session, we assume the network protocol provides loss rates, round-trip times and packet sizes, while the streaming application provides details on the MPEG frame characteristics. The model we develop in the rest of this section allows exploration of the effects of different choices of FEC and temporal scaling will have on the application performance. In particular, the presentation layer can adjust the FEC and temporal scaling patterns so as to optimize the total playable frame rate, which we can compare to typical FEC patterns and video without FEC.

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, Equation 2 provides the probability of successful transmission for each of the frame types:

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

3.3 Playable Frame Rate

Using the TCP-Friendly bandwidth constraint in Equation 1, our model expresses the GOP rate (GOPs per second) analytically (Section 3.3.1). Subsequently, our model computes the playable frame rate using the frame dependency relationships for each of the I, P, and B frame types (Sections 3.3.2-3.3.4). Summing the individual playble frame rates provides the total playable frame rate for the streaming application.

3.3.1 GOP Rate

Given R_F, the target full motion frame rate, the GOP rate (specified in GOPS per second during encoding) is:

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

If, in adapting to the current network bandwidth, 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. Temporal scaling, by discarding frames before transmitting, is used to adapt to current network bandwidth while maintaining 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:

$$R_I = G \cdot q_I \cdot D_I \quad (5)$$

This paper assumes that D_I is always one since I frame is the most important frame in the GOP and no frame will be useful if the I frame is discarded. 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 $R_{P_1} = R_I \cdot q_P \cdot D_{P_1}$. Since each subsequent P_i in the GOP depends upon the success of P_{i-1} and its own successful transmission, we have:

$$R_{P_i} = R_I \cdot {q_P}^i \cdot \prod_{k=1}^{i} D_{P_k} \quad (6)$$

Using the scaling rules in section 2.4, P frames are discarded back to front in the GOP and the P frame playable rate is:

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

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:

$$R_{B_{ij}} = R_{P_{i+1}} \cdot q_B \cdot D_{B_{ij}} \quad when\ 0 \leq i \leq N_P - 1 \quad (8)$$

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:

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

Finally, the playable B frame rate for all B frames is:

$$R_B = \textstyle\sum_{i=0}^{N_P} \sum_{j=0}^{N_{BP}} R_{B_{ij}} \quad (10)$$

3.3.5 Total Playable Frame Rate

The total playable frame rate is the sum of the playable rate for each frame type:

$$R = R_I + R_P + R_B \quad (11)$$

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

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

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 amount of FEC as a function $\mathrm{R}((N_{PD}, N_{BD}), (S_{IF}, S_{PF}, S_{BF}))$. In addition, given RTT and S, the total bandwidth used is also constrained by the TCP-Friendly rate 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, using the equation:

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

Unfortunately, finding a closed form solution for the non-linear function R is difficult due to 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), (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 exhaustively searched to determine the temporal scaling pattern and FEC choices that yield the maximum TCP-Friendly playable frame rate.

4 Analytic Experiments

In this section, we design experiments that use our analytic model to explore the performance of temporarly scaled MPEG video without FEC, with fixed FEC, and with adjusted FEC, where the videos bandwidths are constrained by TCP-friendly data rates.

The MPEG video without FEC has the advantage of not adding overhead to the MPEG data packets, thus using the full available bandwidth to transmit application data, but the disadvantage of being vulnerable to packet loss.

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

The MPEG video with adjusted FEC uses the equations in Section 3 to determine the FEC and temporal scaling patterns that achieve the maximum playable frame rate. This has the advantage of providing the amount of FEC appropriate for the current network conditions, but the disadvantages of not performing well outside of analytic modeling and having a more complex implementation. Section 5 presents experiments conducted to evaluate the effectiveness of the model under more realistic network conditions in and Section 6 considers the complexity of implementation issues.

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

4.1 Methodology

Using the formulas in Section 3, we built a function, `frameRate()`, that takes in application, presentation and network parameters and uses Equation 14 and returns the frame rate. We built a program that, given values of (p, t_{RTT}, S), (N_P, N_B), (S_I, S_P, S_B), and fixed amounts of FEC (S_{IF}, S_{PF}, S_{BF}), uses the `framerate()` function to determine the playable frame rate acheived.

We then built another program that, given values of (p, t_{RTT}, S), (N_P, N_B) and (S_I, S_P, S_B), searches through all combinations of FEC (S_{IF}, S_{PF}, S_{BF}) and temporal scaling patterns (N_{PD}, N_{BD}) using the `frameRate()` function to determine the maximum playable frame rate. The program produces this maximum playable frame rate as well as the adjusted FEC (S_{IF}, S_{PF}, S_{BF}) and temporal scaling (N_{PD}, N_{BD}) required to achieve this optimal rate.

Using these programs, we explore a range of network and application settings, described in Section 4.2. For each set of network and application parmeters, we compare the performance of MPEG video without FEC, MPEG video with fixed FEC, and MPEG video with adjusted FEC.

4.2 System Settings

Table 3 depicts the system parameter settings for the network and application layers. The MPEG frame sizes were chosen based on the mean I, P, B frame sizes measured in [25], and moved up to the nearest integer number of packets. A commonly used MPEG GOP pattern, 'IBBPBBPBBPBB', and a typical full motion frame rate R_F of 30 frames per second were used. The packet size S, round-trip time RTT and packet loss probability p were chosen based on the characteristics of many network connections [29, 3]. For all experiments, the parameters are fixed, except for the packet loss probability, which ranges from 0.01 to 0.04 in steps of 0.001.

Network Layer		Application Layer			
t_{RTT}	50 ms	S_I	25 pkts	N_P	3 fps
S	1 Kbyte	S_P	8 pkts	N_B	8 fps
p	.01 to .04	S_B	3 pkts	R_F	30 fps

Table 3: System Parameter Settings

4.3 Analysis

This section presents the analytical experiment results. We analyze the playable frame rate for non-FEC, fixed FEC and adjusted FEC MPEG video, and explain the effects of FEC and the temporal scaling pattern.

4.3.1 Playable Frame Rate

We compare the playable frame rate for four FEC choices:

1. Fixed FEC (1/0/0): Each I frame receives one 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 [10, 32].

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 application settings in Table 3, this adds approximately 15% overhead for each type of frame, which is typical for many fixed FEC approaches [15, 19, 14, 16].

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.

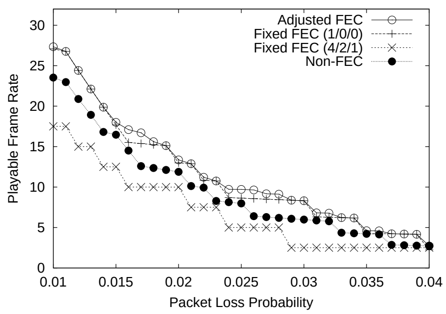
4. Non-FEC: The sender adds no FEC to the video.

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

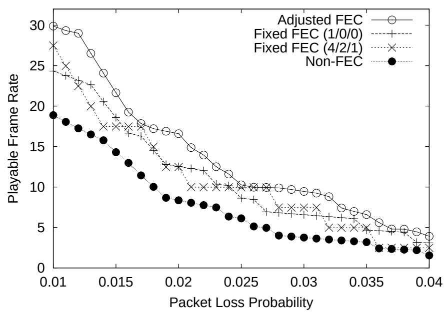
Figure 2 depicts the playable frame rates for each FEC choice. For all figures, the x-axes are the packet loss probabilities, and the y-axes are the playable frame rates. For frame rate targets [30], 24-30 frames per second is full-motion video, 15 frames per second can approximate full motion video for some video content, 7 frames per second appears choppy, and at 3 frames per second or under a 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 is substantial, more than doubling the frame rate at 1% loss, and still maintaining above the minimum 3 fps at 4% loss. The two fixed FEC approaches usually improve playable frame rates over non-FEC video, and FEC(4/2/1) even achieves the playable frame rate provided by adjusted FEC for a few loss rates, such as 2.5%.

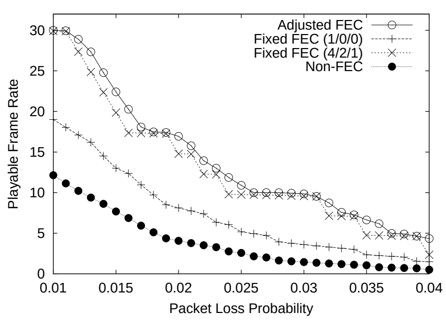
For the smaller video frame sizes in Figure 2a, created by halving the frame sizes in Table 3 and doubling the RTT in order to provide an available bandwidth that allows a visual comparison between graphs, FEC(1/0/0) does substantially better, coming closer to the optimal frame rate achieved by adjusted FEC. FEC(4/2/1) does worse, since the fixed number of FEC packets added is a larger fraction of overhead for the smaller video frames.



a. Small Frame Size



b. Medium Frame Size (as in Table 3)



c. Large Frame Size

Figure 2: Comparision of Playable Frame Rates

For the larger video frame sizes in Figure 2c, created by doubling the frame sizes in Table 3 and halving the RTT, FEC(4/2/1) does substantially better, providing close to the maxium frame rate achieved by adjusted FEC. FEC(1/0/0) does signficantly worse since it does not provide enough protection for the larger frame sizes, with frame rates well below that of adjusted FEC, but still provides improvement over non-FEC.

While there are numerous other fixed FEC and MPEG video choices that could be selected, for space constraints we only present the analysis with above system parameters. However, we feel the common use of these FEC patterns and typical MPEG characteristics justifies their selection. In addition, while other FEC patterns may do as well as adjusted FEC with some MPEG videos under some network conditions, similar to the results in Figure 2, fixed FEC schemes cannot operate effectively over typical ranges of MPEG and network parameters. Additional comparison of adjusted FEC to other fixed FEC schemes can be found in [39].

4.3.2 Adjusting FEC

In an effort to better explain the benefits of adjusted FEC presented in the previous section, we next analyze how FEC is adjusted as loss rates increase.

Figure 3 depicts the breakdown of the adjusted FEC for each I, P, and B frame, yielding the maximum playable frame rate as the loss probability varies. The fixed FEC approaches are not depicted, but they would be represented by horizontal lines since they have 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. Not surprisingly, with adjusted FEC the most important I frames always received more FEC than the P and B frames and the P frames always receive more FEC than the 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 I-frames and then increases it at 1.8%. 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 the third P frame is transmitted (transmitting '`IBBPBBP--P--`'), leaving less leftover data for FEC. At the increased loss rate of 1.8%, the reduced available bandwidth and higher loss rates makes discarding the 3rd P frame (transmitting '`IBBPBBP-----`') and using the remaining bandwidth for FEC the right choice for a higher playable frame rate.

Figure 4 depicts the overhead associated with adjusted FEC compared with Fixed FEC as loss probability varies. The x-axis is the loss probability, and the y-axis is the percentage of overhead computed by taking the number of FEC packets over the MPEG video frame packets transmitted. The overhead of fixed FEC increases as the loss rate increases

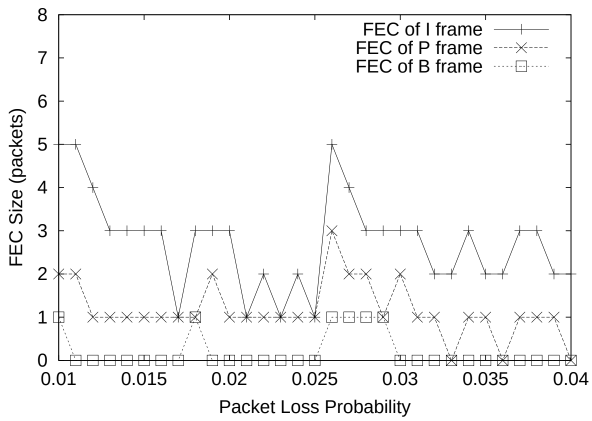


Figure 3: Adjusted FEC Pattern

because the amount of FEC remains fixed, leaving a decreasing amount of bandwidth available for the original video data. The overhead of adjusted FEC also increases, which is appopriate given that the loss rate is increasing thus requiring more FEC to repair losses, but less rapidly than the fixed FEC approaches.

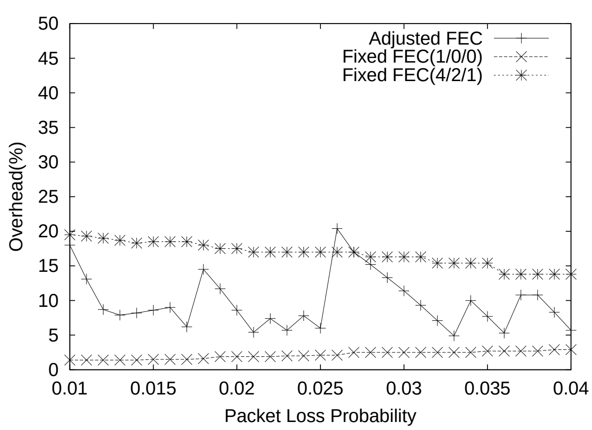
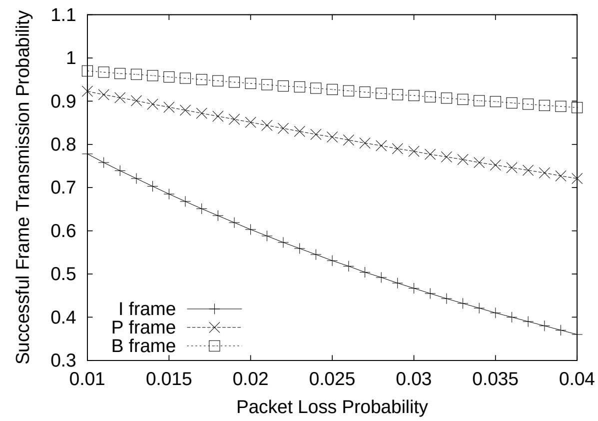
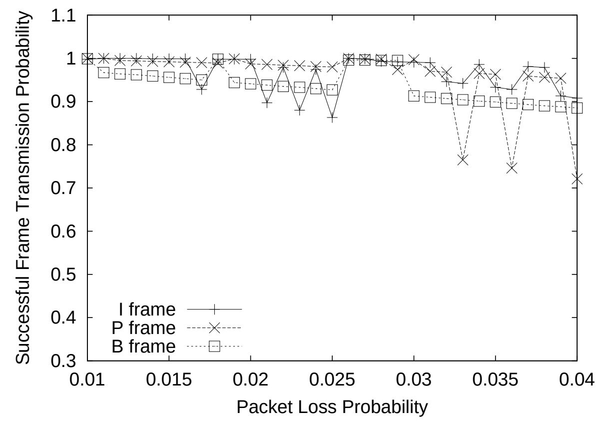


Figure 4: FEC Overhead

The effects of adjusted FEC on the sucessful frame transmission probability can be seen in Figure 5. Without FEC, Figure 5a, the I frames have a decreasing probability of successul transmission. With adjusted FEC, Figure 5b, the most important I frames have the highest transmission probability followed next by the P frames and lastly by the least important B frames.



a. With Non-FEC



b. With Adjusted FEC

Figure 5: Successful Frame Transmission Probability

4.3.3 Temporal Scaling Pattern

Table 4 shows the temporal scaling pattern for adjusted FEC as loss probabilty varies. The '–' symbols denote frames that are discarded by the sender before being transmitted. While p increases, the data rate under the TCP-Friendly constraint decreases. Thus, 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 while the B frames are discarded before the P frames they reference. MPEG video with adjusted FEC must discard slightly more frames than the MPEG video without FEC, but the additional packets saved by the discards are used for FEC. Temporal scaling patterns over a larger range of packet loss probability can be found in [39].

p	Adjusted FEC	Non- FEC
0.010	IBBPBBPBBPBB	IBBPBBPBBPBB
0.015	IBBPBBPB-P--	IBBPBBPBBPB-
0.020	IBBPBBP-----	IBBPBBP-----
0.025	IBBP--P-----	IBBPB-P-----
0.030	IBBP--------	IBBP--------
0.035	IB-P--------	IBBP--------
0.040	I--P--------	I--P--------

Table 4: Temporal Scaling Pattern

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 applications and network conditions. For the experiments in Setion 4, the application layer and network layer parameters remained fixed for the duration of each video run. This simplified environment allowed us to use of our model to clearly illustrate the effects of adjusted FEC over that of fixed FEC and non-FEC approaches. However, in practice, MPEG video frame sizes change over time, even 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 the model's accuracy in predicting playable frame rate in simulations that characterize more realistic network conditions and thereby to determine the effectiveness of our streaming protocol in real Internet situations. Specifically:

1. The analytic experiments assumed an accurate estimate of the packet loss probability from the network protocol. In Section 5.1, the effects of error in the packet loss estimate on the model's predictive quality are considered.

2. Section5.2 introduces bursty packet losses derived from previous Internet streaming measurements to determine the impact of the independent packet loss assumption on the model's accuracy.

3. The analytic experiments assumed the round-trip time was fixed for the life of the flow. In Section 5.3, we use our model to determine the appropriate temporal scaling assuming fixed round-trip times and then apply more realistic round-trip times obtained from traces from previous Internet streaming experiments.

4. The analytic experiments assumed the size of each type of I, P and B frame was constant over the entire video. In Section 5.4, we use our model assuming a fixed frame size and then apply 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 simulation. 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 a comparison of the playable frame rate without FEC indicates 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 3.

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. The possible penalty for under-predicting the loss rate is to produce too little FEC for effective repair. The possible penalty for over-prediciting the loss rate is to produce more FEC than is needed for effective repair, thus leaving less bandwidth for the MPEG data. Three sets of simulation experiments with different 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%, the average margin for error reported based on numerous simulations in [13].

2. The actual loss rate was double the predicted loss rate.

3. The actual loss rate was half the predicted loss rate.

For each loss case, we used the predicted loss rate p 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 above actual losses and measured the actual playable frame rate at the receiver.

Figure 6 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 under-estimated, 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 sub-optimal use of FEC. However, the actual frame rates achieved differ by very little. Moreover, for the practical loss prediction errors of 0.006, the actual frame rates are nearly identical to the predicted frame rates, suggesting using our model to determine proper FEC and temporal scaling can be effective in practice.

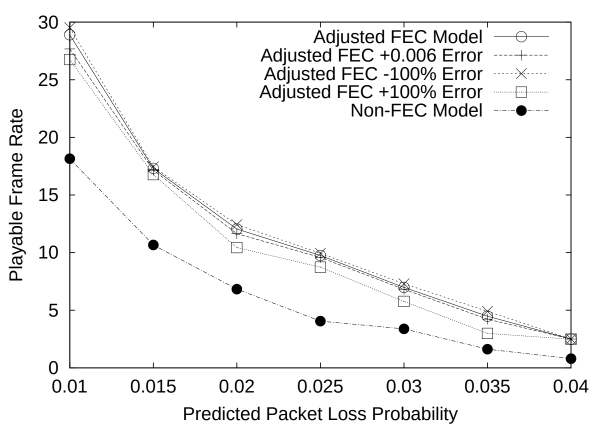
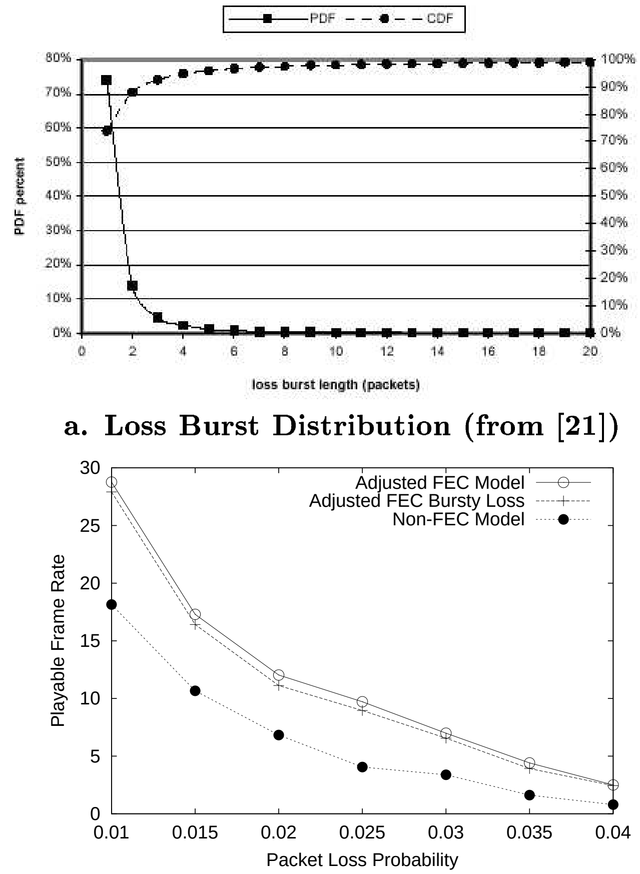


Figure 6: Impact of Inaccurate Loss Prediction

5.2 Bursty Loss

In our analytic model, we assume loss events are independent, while in practice, losses have been shown to be bursty [20, 29]. The possible impact of bursty losses is that Section 2.2, it becomes less likely that K packets of the N total frame packets can be recovered, resulting in a decreased playable frame rate.

A series of traces from a previous Internet measurement study [9] were used 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 [21] and depicted in Figure 7a, to provide bursty loss events.



a. Loss Burst Distribution (from [21])

b. Playable Frame Rate

Figure 7: Impact of Bursty Loss

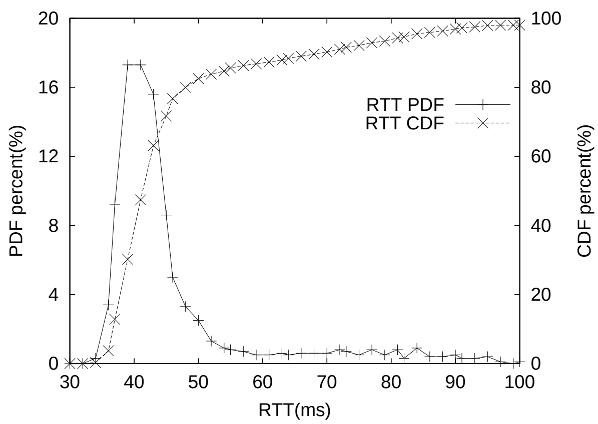
We used our model to determine the adjusted FEC and predicted frame rate assuming independent losses. Then, we simulated streaming the MPEG video using the trace driven loss events and loss bursts and measured the actual playable frame rate at the receiver.

Figure 7 depicts the playable frame rates for the simulations along with the playable frame rates estimated by our model. The bursty packet losses do result in the adjusted FEC being less effective, but only marginally, suggesting using our model to determine FEC based on independent losses yields good performance in practice.

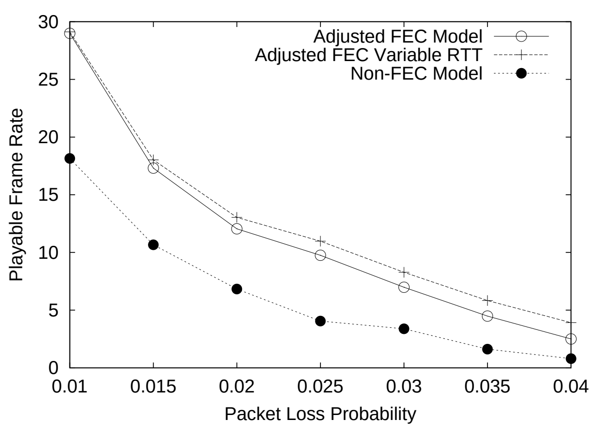
5.3 Variable Round-Trip Times

In our analytical model, we assume round-trip times (RTTs) are fixed for the entire flow, while in practice, RTTs can vary considerably. The possible impact of variable RTTs is that the bandwidth estimate by using the fixed average RTT is inaccurate, therefore making the temporal scaling and amount of FEC less effective.

To simulate the effects of variable round-trip times, we selected a trace from [9], depicted in Figure 8a, that had a median RTT of about 45 ms.



a. RTT Distribution (from [9])



b. Playable Frame Rate

Figure 8: Impact of Variable RTT

We used our model to determine the adjusted FEC and temporal scaling patterns assuming a fixed RTT of 45 ms. Then, we simulated streaming the MPEG video using the RTT trace and measured the actual frame playout rate at the receiver.

Figure 8b depicts the playable frame rates for the simulations along with the playable frame rates estimated by our model. Perhaps somewhat surprisingly, the variable RTT curve has a slightly higher playable frame rate than our model estimated in using the average RTT. We attribute this to the fact that the RTT distribution we selected is not 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, suggesting our model will be effective in practice.

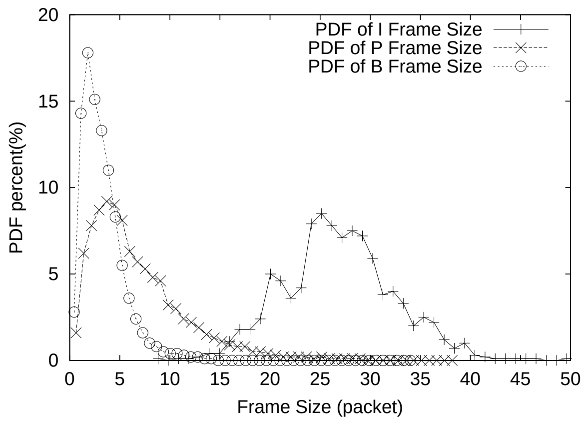
5.4 Variable MPEG Frame Sizes

In our analytic model, we assume the MPEG frame sizes are fixed for the entire video. In practice, MPEG frame sizes change constantly, even inside a GOP. There are two possible impacts of variable sized frames: 1) the adjusted FEC chosen for the fixed average frame sizes will be inappropriate for the actual frame sizes size, resulting in a lower playable frame rate; 2) our model will have to be re-applied for each GOP to chose the appropriate FEC adjustment, thus increasing the overhead.

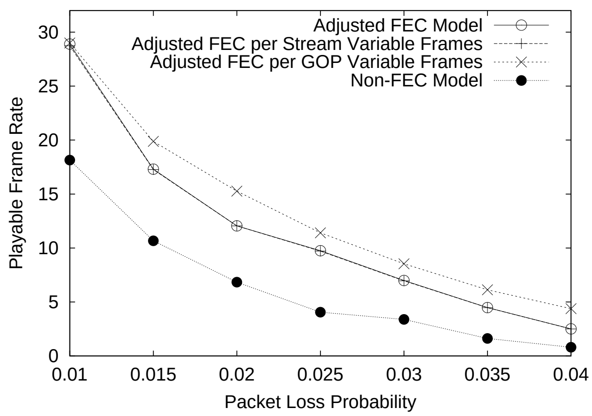
To simulate the effects of variable MPEG frame sizes, we selected a frame size trace from [34], depicted in Figure 9a, with the average frame sizes S_I, S_P and S_B of 28, 8 and 4 packets respectively.

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

Figure 8b depicts the playable frame rates for the simulations along with the playable frame rates estimated by our model. The frame rate depicted by the model is slighly higher than the actual frame rate achieved, however, the difference is slight. There is almost no difference in the FEC that is adjusted each GOP compared with the FEC that is adjusted based on the average frame sizes, which suggests there is little to be gained in terms of playable frame rate by adjusting FEC each GOP.



a. Frame Sizes Distributions (from [34]



b. Playable Frame Rate

Figure 9: Impact of Variable Frame Size

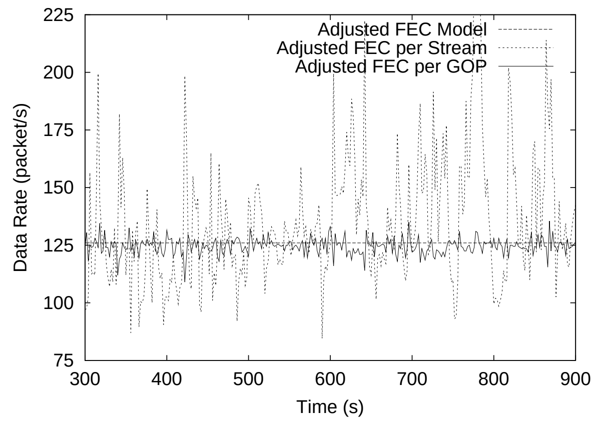


Figure 10: Data Rates

However, Figure 10 depicts the data rates under 2.5% loss for the simulated FEC scheme compared the the data rates predicted by the model. The model, which uses a fixed frame size, predicts a constant data rate. The adjusted FEC applied for the average frame size, while still TCP-friendly over long time periods, has considerable variation in its data rate. The adjusted FEC applied to each GOP, however, has a much smoother data rate, significantly closer to the predicted, constant data rate. Overall, smooth data rates are much easier for networks to manage than bursty data rates.

6 Discussion

While our work thus far has applied our model only in analytic and simulation experiments, the goal is to use our model as the core of video presentation layer that does temporal scaling with adaptive FEC. This presentation layer would not require changes to an MPEG codec or MPEG streaming server. The presentation layer could monitor and measure the frame types as they were transmitted (for live content) or ahead of time (for stored content), and either use our analytic model to apply FEC and temporal scaling based on the average frame sizes or based on the frame sizes for the current GOP. The presentation layer would require a TCP-friendly network transport protocol, such as [4, 13, 31, 33], where the transport protcol API would need to provide the loss probability p, round-trip time t_{RTT}, and the packet size S.

Adjusting the FEC and temporal scaling patterns can be done in software in real time even with an exhaustive search of all combinations. Using our model to find the best adjusted FEC and temporal scaling pattern for the GOP of '`IBBPBBPBBPBB`' takes about 30 ms on a PIII 800 MHz, which is much less than the real-time playout time of 400 ms for the GOP. Moreover, our current implementation is not optimized, so it is likely the processing time can be substantially reduced. For streaming devices that are have limited processing power, the appropriate FEC and temporal scaling patterns can be stored in a lookup table ahead of time, avoiding per GOP processing.

While the benefits of adjusting FEC each GOP are not substantial compared with applying an adjusted FEC pattern over the whole video, the benefits of the smoother network data rate are not trivial. Burst network traffic can cause buffer overflows, leading to increased loss rates, decreased data rates and leading to the need for more FEC and less MPEG data. Worse, for higher loss rates, the increased FEC will cause even higher fluctations in the data rates over time, potentially causing even more network instability. By adjusting FEC each GOP, basically the video is temporally scaled to fit under the TCP friendly rate while the leftover bandwidth can be used for FEC, thus creating a very smooth stream, helping to stabilize network conditions.

7 Conclusions

This paper proposes an analytic model that fully 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. Using our model, we can determine the optimal adjustment of FEC and temporal scaling, accounting for both the network conditions and application settings.

Our analytic experiments with our model show that adjusting FEC and temporal scaling shows significant advantages to current approaches. Adjusted FEC 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 range of MPEG encoding and network conditions. Our simulation experiments show using our model is practical over a range of realistic system conditions: inaccurate loss predictions, bursty packet loss, variable round-trip time, and variable MPEG frame size. 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, providing substantial increases in playable frame rates while maintaining TCP friendly bandwidth.

Possible future work includes integrating our model with the implementation of TCP friendly network protocol and MPEG streaming system. Other possible future work includes extending our model to analyze other types of media repair, such as media-dependent FEC as in [19] and selective retransmissions as in [10].

References

[1] S. Acharya and B. Smith. An Experiment to Characterize Videos Stored on the Web. In *Proceedings of MMCN*, January 1998.

[2] A. Albanese, J. Bomer, J. Edmonds, M. Luby, and M. Sudan. Priority Encoding Transmission. *IEEE Transactions on Information Theory*, 46(6):1737 – 1744, 1996.

[3] M. Allman. A Web Server's View of the Transport Layer. *ACM Computer Communication Review*, 30(4), Oct. 2000.

[4] H. Balakrishnan, H. Rahul, and S. Seshan. An Integrated Congestion Management Architecture for Internet Hosts. In *Proceedings of ACM SIGCOMM*, Sept. 1999.

[5] P. Bocheck, A. Campbell, S.-F. Chang, and R. Lio. 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)*, June 1999.

[6] J.-C. Bolot, S. Fosse-Parisis, and D. Towsley. Adaptive FEC-Based Error Control for Internet Telephony. In *Proceedings of IEEE INFOCOM*, Mar. 1999.

[7] J. Boyce and R. Gaglianello. Packet loss effects on mpeg video sent over the public internet. In *Proceedings of ACM Multimedia*, pages 181–190, Bristol, U.K., Sept. 1998.

[8] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, S. F. D. Estrin, V. Jacobson, G. Minshall, C. Partridge, K. R. L. Peterson, S. Shenker, J. Wroclawski, and L. Zhang. Recommendations on Queue Management and Congestion Avoidance in the Internet. *IETF Request for Comments (RFC) 2309*, Apr. 1998.

[9] J. Chung, M. Claypool, and Y. Zhu. Measurement of the Congestion Responsiveness of RealPlayer Streaming Video Over UDP. In *Proceedings of the Packet Video Workshop (PV)*, April 2003.

[10] N. Feamster and H. Balakrishnan. Packet Loss Recovery for Streaming Video. In *12th International Packet Video Workshop*, Apr. 2002.

[11] W. Feng, D. Kandlur, D. Saha, and K. Shin. Stochastic Fair Blue: A Queue Management Algorithm for Enforcing Fairness. In *Proceedings of IEEE INFOCOM*, Apr. 2001.

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

[13] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-Based Congestion Control for Unicast Applications. In *Proceedings of ACM SIGCOMM Conference*, pages 45 – 58, 2000.

[14] S. Gringeri, G. Khasnabish, A. Lewis, K. Shuaib, R. Egorov, and B. Basch. Transmission of MPEG-2 Video Streams over ATM. *IEEE Multimedia*, 5(1):58 – 71, Jan-Mar 1998.

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

[16] F. Hartanto and H. R. Sirisena. Hybrid Error Control Mechanism for Video Transmission in the Wireless IP Networks. In *Proceedings of the Tenth IEEE Workshop on Local and Metropolitan Area Networks (LANMAN)*, Nov. 1999.

[17] M. Hemy, U. Hangartner, P. Steenkiste, and T. Gross. MPEG System Streams in Best-Effort Networks. In *Proceedings of Packet Video Workshop*, Apr. 1999.

[18] 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.

[19] Y. Liu and M. Claypool. Using Redundancy to Repair Video Damaged by Network Data Loss. In *Proceedings of IS&T/SPIE/ACM Multimedia Computing and Networking (MMCN)*, Jan. 2000.

[20] D. Loguinov and H. Radha. Measurement Study of Low-bitrate Internet Video Streaming. In *In Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, Nov. 2001.

[21] D. Loguinov and H. Radha. Measurement Study of Low-bitrate Internet Video Streaming. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop (IMW)*, November 2002.

[22] R. Mahajan, S. Floyd, and D. Wetherall. Controlling High-Bandwidth Flows at the Congested Router. In *Proceedings of the 9th International Conference on Network Protocols (ICNP)*, Nov. 2001.

[23] A. Mena and J. Heidemann. An Empirical Study of Real Audio Traffic. In *In Proceedings of the IEEE Infocom*, pages 101 – 110, Mar. 2000.

[24] J. Mitchell and W. Pennebaker. *MPEG Video: Compression Standard*. Chapman and Hall, 1996. ISBN 0412087715.

[25] M.Krunz, R.Sass, and H.Hughes. Statistical Characteristics and Multiplexing of MPEG Streams. In *Proceedings of IEEE Infocom*, pages 455–462, April 1995.

[26] C. Padhye, K. Christensen, and W. Moreno. A New Adaptive FEC Loss Control Algorithm for Voice Over IP Applications. In *Proceedings of IEEE International Performance, Computing and Communication Conference*, Feb. 2000.

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

[28] K. Park and W. Wang. QoS-Sensitive Transport of Real-Time MPEG Video Using Adaptive Forward Error Correction. In *Proceedings of IEEE Multimedia Systems*, pages 426 – 432, June 1999.

[29] V. Paxson. End-to-End Internet Packet Dynamics. *IEEE/ACM Transactions on Networking*, Fall 1999.

[30] Real Networks Incorporated. RealProducer User's Guide, copyright 2000. URL: http://www.service.real.com/help/library-/guides/producerplus85/producer.htm.

[31] R. Rejaie, M. Handley, and D. Estrin. RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet. In *Proceedings of IEEE Infocom*, 1999.

[32] I. Rhee. Error Control Techniques for Interactive Low-Bit Rate Video Transmission over the Internet. In *Proceedings of ACM SIGCOMM Conference*, Sept. 1998.

[33] I. Rhee, V. Ozdemir, and Y. Yi. TEAR: TCP Emulation at Receivers – Flow Control for Multimedia Streaming. Technical report, Department of Computer Science, NCSU, 2000.

[34] O. Rose. Statistical Properties of MPEG Video Traffic and their Impact on Traffic Modeling in ATM Systems. In *Proceedings of 20th Conference of Local Computer Networks*, pages 397–406, Oct 1995.

[35] T.Nguyen and A.Zakhor. Distributed Video Streaming with Forward Error Correction. In *12th International Packet Video Workshop*, April 2002.

[36] A. Tripathi and M. Claypool. Improving Multimedia Streaming with Content-Aware Video Scaling. In *Workshop on Intelligent Multimedia Computing and Networking (IMMCN)*, Mar. 2002.

[37] Y. Wang, M. Claypool, and Z. Zuo. An Empirical Study of RealVideo Performance Across the Internet. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, Nov. 2001.

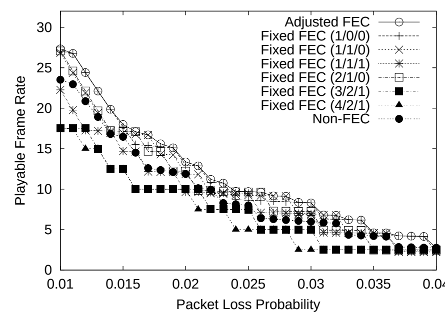
[38] H. Wu, M. Claypool, and R. Kinicki. Adjusting Forward Error Correction for TCP-Friendly Streaming MPEG. Technical Report WPI-CS-TR-02-29, CS Department, Worcester Polytechnic Institute, Dec. 2002.

[39] H. Wu, M. Claypool, and R. Kinicki. Adjusting Forward Error Correction with Temporal Scaling for TCP-Friendly Streaming MPEG. Technical Report WPI-CS-TR-03-10, CS Department, Worcester Polytechnic Institute, Apr. 2003.

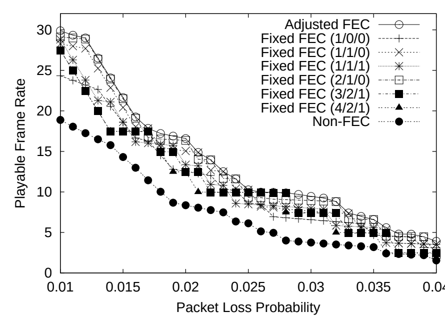
8 Appendix

8.1 Detailful Comparision of Playable Frame Rates

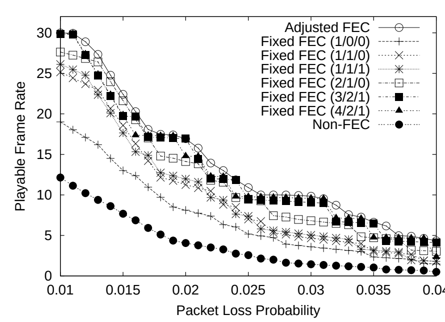
Figure 11 gives a detailful comparision of Playable Frame Rates between Adjusting FEC and Fixed FECs.



a. Small Frame Size



b. Medium Frame Size (as in Table 3)



c. Large Frame Size

Figure 11: Comparision of Playable Frame Rates

8.2 Detailful Temporal Scaling Patterns

Table 5 shows the detailful Temporal Scaling patterns with different loss probabilities.

p	Adjusted FEC	Non- FEC
0.010	IBBPBBPBBPBB	IBBPBBPBBPBB
0.011	IBBPBBPBBPBB	IBBPBBPBBPBB
0.012	IBBPBBPBBPBB	IBBPBBPBBPBB
0.013	IBBPBBPBBPB-	IBBPBBPBBPBB
0.014	IBBPBBPBBP--	IBBPBBPBBPBB
0.015	IBBPBBPB-P--	IBBPBBPBBPB-
0.016	IBBPBBP--P--	IBBPBBPBBP--
0.017	IBBPBBP--P--	IBBPBBPB-P--
0.018	IBBPBBP-----	IBBPBBP--P--
0.019	IBBPBBP-----	IBBPBBP-----
0.020	IBBPBBP-----	IBBPBBP-----
0.021	IBBPBBP-----	IBBPBBP-----
0.022	IBBPB-P-----	IBBPBBP-----
0.023	IBBPB-P-----	IBBPBBP-----
0.024	IBBP--P-----	IBBPB-P-----
0.025	IBBP--P-----	IBBPB-P-----
0.026	IBBP--------	IBBP--P-----
0.027	IBBP--------	IBBP--P-----
0.028	IBBP--------	IBBP--------
0.029	IBBP--------	IBBP--------
0.030	IBBP--------	IBBP--------
0.031	IBBP--------	IBBP--------
0.032	IBBP--------	IBBP--------
0.033	IBBP--------	IBBP--------
0.034	IB-P--------	IBBP--------
0.035	IB-P--------	IBBP--------
0.036	IB-P--------	IB-P--------
0.037	I--P--------	IB-P--------
0.038	I--P--------	IB-P--------
0.039	I--P--------	IB-P--------
0.040	I--P--------	I--P--------

Table 5: Temporal Scaling Pattern