A Model for MPEG with Forward Error Correction and TCP-Friendly Bandwidth

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

ABSTRACT

The growing requirement of TCP-Friendly bandwidth use by streaming video plus the proven advantages of Forward Error Correction (FEC) to combat packet loss presents the opportunity to optimize the amount of FEC in a TCP-Friendly video stream. In this paper, we derive an analytical model for predicting the playable frame rate in a TCP-Friendly MPEG stream with FEC. Our model characterizes the Group Of Pictures (GOP) and Forward Error Correction (FEC) that are part of the MPEG video transmission. Assuming a network estimate for the packet loss probability, our model incorporates TCP-Friendly throughput constraints to calculate a total playable frame rate. For a given packet loss probability, we use our model to search the variable space to find the MPEG configuration that yields the optimal playable frame rate. Analysis over a range of network conditions indicates that adjusting FEC can provide a significant performance improvement, while adjusting a well-chosen GOP will contribute little improvement. Further analysis shows that a poor choice for a GOP can result in a large degradation of the playable frame rate. Overall, by introducing moderate amounts of FEC overhead, frame rates can be improved 10 to 50 times under network conditions with moderate to high loss rates.

Categories and Subject Descriptors

C.2.m [**Computer-Communication Networks**]: Miscellaneous

General Terms

Performance, Design

Keywords

Multimedia Networking, MPEG, Forward Error Correction, TCP-Friendly

NOSSDAV'03, June 1–3, 2003, Monterey, California, USA.

1. INTRODUCTION

The growing number of users with high bandwidth Internet connections and the increasing power of desktop computers have fueled the use of the Internet to carry potentially high-quality video. Increasingly, Web sites are offering streaming video clips of news broadcasts, music television and live sporting events. Users can access these streaming video clips through a Web browser by simply clicking on a link and having the Web browser start up an associated video player.

While TCP is the de facto standard transport protocol for typical Internet applications, there are as of yet no widely accepted transport protocols for streaming media applications. Unlike typical Internet traffic, streaming video is sensitive to delay and jitter, but can tolerate some data loss. Additionally, streaming video transmissions tend to yield better quality video when the underlying protocol provides a steady data rate rather than the bursty data rate often associated with window-based network protocols. For these reasons, streaming video applications often use UDP as a transport protocol rather than TCP, in order to give the application control over the packets sent.

Although UDP can provide rate-based control over a video stream, it is an unresponsive protocol in that it does not reduce its data rate when an Internet router drops packets to indicate congestion. Since the number of streaming flows on the Internet is expected to grow rapidly, using UDP as the underlying transport protocol for streaming video flows means that UDP will both introduce traffic to cause congestion and not respond to relieve the saturation of a bottleneck due to the congestion. As some researchers have argued [9], response to congestion is critical to the health of the Internet. To avoid this impending congestion collapse, some Internet researchers have argued that all Internet flows should be required to be *TCP-Friendly* [5, 26]. A flow is TCP-Friendly if its data rate does not exceed the maximum arrival of a conformant TCP connection in the same network conditions. Recent research has proposed rate-based TCP-Friendly protocols for streaming media [10, 24, 25] as alternatives to UDP.

Besides the problem of avoiding congestion, video streams must react to the problem of packet loss. While streaming video applications can tolerate some data loss, too much data loss can produce unacceptable media quality. Moreover, the intra-frame dependencies needed to achieve high-compression rates in video exacerbate the degradation in quality when primary frames are lost. While retransmissions are an appropriate response to packet loss for most applications, even many streaming applications [8], large timeout periods cause problems for applications such as video conferences that have short end-to-end delay requirements.

In an effort to improve MPEG [17] video quality in the presence of network congestion while reducing the data rate, [6] and [27] vary the size and makeup of the Group of Pictures (GOP) to reduce network bandwidth during congestion. However, these studies do not address how to choose the best GOP pattern to use with MPEG, nor are they explicitly TCP-Friendly. Thus, the question remains whether there exists an optimal GOP pattern given TCP-Friendly constraints for a streaming flow's network loss rate.

By adding redundant data to a media stream [4, 12, 18, 19, 21], Forward Error Correction (FEC) can be used to repair the damage to the media stream due to packet loss. Used properly, FEC can reduce or eliminate packet loss and partially or fully insulate video applications from degraded quality [14]. However, FEC adds overhead to the video stream. Hence, if a video flow is to operate within the TCP-Friendly throughput constraints, adding FEC will reduce the playable rate of the transmitted video content. Thus, given an estimated network loss rate and the constraint of a TCP-Friendly data rate, the choice of FEC can be cast as a constrained optimization problem where the goal is to choose the combination of FEC packets that optimizes the quality of the video stream. Current works use either apriori, static FEC choices [2, 11] or adapt FEC to perceived packet loss on the network without regard to TCP-Friendly data rate constraints [4, 19, 21].

This paper builds upon the work of Mayer et al [16] to derive an analytical model for streaming video with FEC in a TCP-Friendly environment. Given an estimate of the loss rate that a streaming video flow experiences, our model uses GOP patterns and FEC weighted by frame type to optimize the total playable frame rate. We design experiments to search the space of possible GOP and FEC combinations to find the maximum playable frame rate given a TCP-Friendly bandwidth constraint. Over a range of network and video conditions, our results show that selecting the optimal GOP pattern over a set of typical GOP patterns has only moderate benefit while adjusting FEC to different packet loss levels provides significant benefit to the playable frame rate and video quality.

The remainder of the paper is organized as follows: Section 2 provides background knowledge to the work in this paper; Section 3 introduces our analytic model; Section 4 discusses our methodology and experimental settings; Section 5 analyzes the results; and Section 6 summarizes the paper and presents possible future work.

2. BACKGROUND

This section briefly reviews topics and terminology that are the foundation for the analytic model developed Section 3.

2.1 TCP-Friendly

A flow is considered to be TCP-Friendly if its bandwidth usage, in the presence of a constant loss rate, is less than the equivalent TCP flow. Padhye et al [20] analytically derived the following TCP equation:

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

where s is the packet size, t_{RTT} is the round-trip time, p is the packet loss probability (from 0 to 1), and t_{RTO} is the TCP retransmit timeout value. Thus, T provides an upper bound on the TCP-Friendly sending rate. We assume that streaming flows are TCP-Friendly, which could be satisfied by using a TCP-Friendly flow such as [10, 24, 25].

2.2 Forward Error Correction

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 [23] 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 success rate of FEC on application layer frames, we model the sending of packets as a series of Bernoulli trials. Thus the probability $q(N,K,p)$ that a K-packet data frame is successfully transmitted with $N-K$ redundant FEC packets in a lossy network 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 5.4.

2.3 MPEG

MPEG (Motion Picture Expert Group) [17], a popular standard for compressing video streams, uses both intra-frame and inter-frame compression. MPEG I (intra-coded) frames are encoded independently and focus on encoding similarities within a video scene. MPEG P (predictive-coded) frames are encoded using predictions from preceding I or P frame in the video sequence. MPEG B (bi-directionally predictive-coded) frames are encoded using predictions from the preceding and succeeding I or P frames.

MPEG-encoded video typically repeats the pattern of I, P, and B frames (known as a Group of Pictures or GOP) for the duration of an individual video stream. Figure 1 shows a sample GOP, where the second I frame in the figure marks the beginning of the next GOP. The arrows indicate frame dependency relationships. Note that the loss of one P frame can render some of other P and B frames undecodable, and the loss of one I frame can result in the whole GOP being undecodable. This implies that I frames are more important than P frames, and P frames are more important than B frames.

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[1]. Obviously, $N_B = (1+N_P)\times N_{BP}$. Using

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

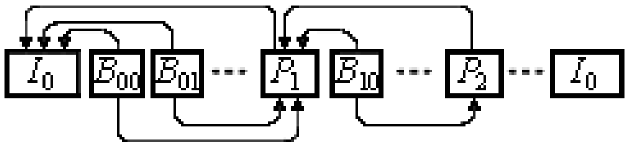


Figure 1: A sample MPEG Group Of Pictures

these terms to characterize GOP patterns, a specific GOP pattern can be identified uniquely by G(N_P,N_{BP}). For example, GOP(2,2) signifies the GOP pattern 'IBBPBBPBB'.

As in Figure 1, for the remainder of the paper, we will use subscripts 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 $1 \le i \le N_P$, and B frames are expressed as B_{ij}, where $0 \le i \le N_P$ and $0 \le j < N_{BP}$. For example, P_i is the $(i-1)$th P frame, and B_{ij} is the jth B frame in the ith interval of I and P frames.

3. ANALYTICAL MODEL

This section provides the details of the analytic model we develop to investigate means to improve the playable frame rate of TCP-Friendly streaming video flows in the presence of network packet loss. First, we identify system parameters related to TCP-Friendly MPEG flows (see Section 3.1). Next, working from frame size parameters and representations of the amount of FEC to add per frame type, we create a system of equations to characterize the probability of a successful transmission for each MPEG frame type (see Section 3.2). Lastly, by adding information about the Group Of Picture (GOP) structure and frame dependencies, we derive formulas for flow transmission rate and playable frame rate (see Section 3.3).

3.1 Parameters and Variables

The following six parameters are treated as fixed attributes of a streaming MPEG flow for the duration of the flow's lifetime:

t_{RTT}: the round-trip time
t_{RTO}: the TCP retransmission timeout interval
s: the packet size (in bytes)
S_I: the size of I frames (in packets)
S_P: the size of P frames (in packets)
S_B: the size of B frames (in packets)

The remaining six attributes of an MPEG video flow over a lossy network are treated as variables for the remainder of our discussion:

p: the packet loss percentage
N_P: the number of P frames in a GOP
N_{BP}: the number of B frames in an interval of I and P frames
S_{IF}: the number of FEC packets added to each I frame
S_{PF}: the number of FEC packets added to each P frame
S_{BF}: the number of FEC packets added to each B frame

p is estimated by the network protocol, whereas N_P and N_{BP} can be modified in the MPEG encoding to give an adjusted GOP, and S_{IF},S_{PF} and S_{BF} can be modified to provide an adjusted FEC.

The strategy in the model is to assume that the network is able to provide an estimate of the current network loss probability and round-trip time while the MPEG application can provide details on the video characteristics. Armed with this information, the model can be used to adjust the other five attributes (N_P, N_{BP}, S_{IF}, S_{PF} and S_{BF}) so as to optimize the playable frame rate and explore the effect of today's typically non-adjusted MPEG video.

3.2 Successful Frame Transmission Probabilities

Given specific I, P, and B frame sizes, a GOP, and the distribution of redundant FEC packets added to each frame type, Equation 2 can be used to compute the probabilities of successful transmission for each frame type as following:

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

3.3 Playable Frame Rate

At this point the TCP-Friendly behavior is imposed via the throughput constraints expressed in Equation 1. With the five system attributes introduced as variables, the GOP rate (GOPs per second) can be expressed analytically because the total size of one GOP is known. Subsequently, the frame dependency relationships for I, P, and B frames are used to compute the playable rate for each frame type.

3.3.1 GOP Rate

Assuming Equation 1 provides an expression for T, the GOP rate is computed as:

$$G = \frac{T/s}{(S_I + S_{IF}) + N_P(S_P + S_{PF}) + N_B(S_B + S_{BF})} \tag{4}$$

where $N_B = (1 + N_P) \times N_{BP}$.

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. With only one I frame per GOP, the playable I frame rate is simply:

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

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 $R_{P_1} = R_I \cdot q_P$. Since each subsequent P_i in the GOP depends upon the success of P_{i-1} and its own successful transmission, we have by induction:

$$R_{P_i} = R_I \cdot {q_P}^i \tag{6}$$

and the playable P frame rate for all P frames:

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

3.3.4 Playable Rate of B Frames

All B frames in the same interval between an I or P frame have the same dependency relationship and thus these B frames all have the same playable frame rate.

A B frame that precedes a P 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 effects have already been accounted for in the success probability of this P frame. Thus:

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

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

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

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

$$\begin{aligned} R_B &= N_{BP} \cdot \textstyle\sum_{i=0}^{N_P} R_{B_{i0}} \\ &= N_{BP} \cdot G \cdot q_I \cdot q_B \cdot \left(\frac{q_P - q_P^{N_P+1}}{1-q_P} + q_I \cdot q_P^{N_P}\right) \end{aligned} \tag{10}$$

3.3.5 Total Playable Frame Rate

The total playable frame rate is:

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

Using the above equations for R_I, R_P and R_B, the total playable frame rate is:

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

4. EXPERIMENTS

In this section, we use the closed form formula for the total playable frame rate derived in Section 3 to explore the space of GOP and FEC choices given TCP-Friendly bandwidth constraints. We use experimental instances of the probabilistic model to exhaustively search the system variable space to find the maximum playable frame rate for a given packet loss percentage.

4.1 Methodology

To study the effects of adjusted GOP and FEC on performance for a given loss percentage, p, we execute the following steps of a model to optimize the playable frame rate:

1. Given a specified p, determine the TCP-Friendly data rate T from Equation 1.

2. For each set of the other five system variables, compute the playable frame rate R using Equations 4 and 12. We characterize the variation of these five variables in terms of two searching dimensions: the GOP dimension and the FEC dimension. For different fixed combinations of these dimensions, we compute the rate R for each possible set of the remaining variables. Once the maximum playable frame rate is determined, we record the value of each of the five variable settings to produce this rate.

3. For all the combinations of searching dimensions, we compare the maximum rates to consider the benefits of adjustments in both the GOP and FEC dimensions. Later in the analysis, we run experiments with the model to consider the behavior of the optimal pattern of GOP and FEC in the face of different loss percentages.

4.2 System Settings

Table 1 depicts our network settings, with packet size, RTT and loss characteristics of many typical network connections [3, 22], and $t_{RTO} = 4t_{RTT}$ as in [10].

t_{RTT}(ms)	50
t_{RTO}(ms)	200
s (KB)	1
p (%)	0.5, 1, 2, ... 10

Table 1: Network settings

Table 2 lists the MPEG frame sizes used for the experiments. These sizes were chosen based on the mean I, P, and B frame sizes measured by Krunz et al [13], and moved up to the next integer to model frames with an integer number of fixed packets.

	I Frames	P Frames	B Frames
Mean Size	24.64KB	7.25KB	2.45KB
Number of Packets	25	8	3

Table 2: MPEG frame size

When implementing our model in a real network, the network parameters (p, t_{RTT} and t_{RTO}) would be provided by the transport protocol as in [10]. The MPEG frame sizes can would be provided by the MPEG application after encoding.

4.3 Searching Dimensions

The search choices for optimizing the playable frame rate are viewed as moving in two dimensions - one for FEC components and the other for GOP patterns. The combination of choices for N_P and N_{BP} determines the GOP structure and represents the GOP dimension. Dividing redundant FEC packets into S_{IF}, S_{PF}, and S_{BF} packets indicates the relative importance of these frame types and represents the FEC dimension.

Table 3 lists the four possible combinations of the two searching dimensions explored in Section 5. The word 'Adjusted' implies the graphs in Section 5 show the optimal playable rate over the search space dimension. Schema A is called 'Fixed' since it uses no FEC and a fixed GOP, GOP(4,2) ('IBBPBBPBBPBBPBB'), a common used GOP [1, 16]. Schema B is called 'Adjusted GOP' since it has no FEC and adjusts N_P and N_{BP} to optimize the playable frame rate. Schema C is called 'Adjusted FEC' since it adjusts S_{IF}, S_{PF} and S_{BF} with a fixed GOP, GOP(4,2). Schema D adjusts all of N_P, N_{BP}, S_{IF},S_{PF} and S_{BF}.

A few reasonable constraints were imposed to reduce the time required to search the variable space. The maximum frame rate allowed is 30 frames per second, typical of full-motion video. The maximum size in frames of the GOP is 15, as recommended in [17], and the maximum number of B

Schema	Description
A	Fixed
B	Adjusted GOP
C	Adjusted FEC
D	Adjusted FEC and GOP

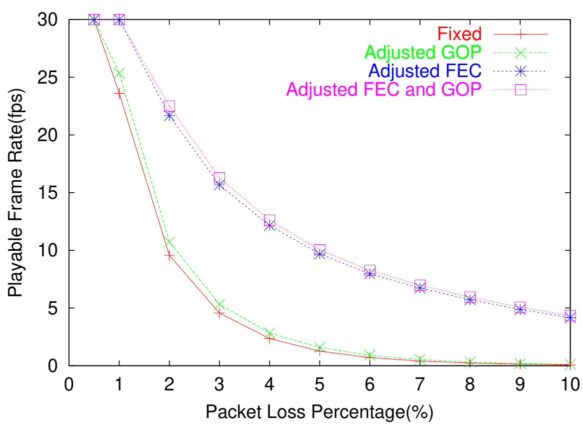
Table 3: Searching dimension schemas

frames in the I-P frame intervals is 3. Although the search is exhaustive, using the analytic model to find the best FEC and GOP pattern only takes about 30 ms on a P-3 800 MHz machine.

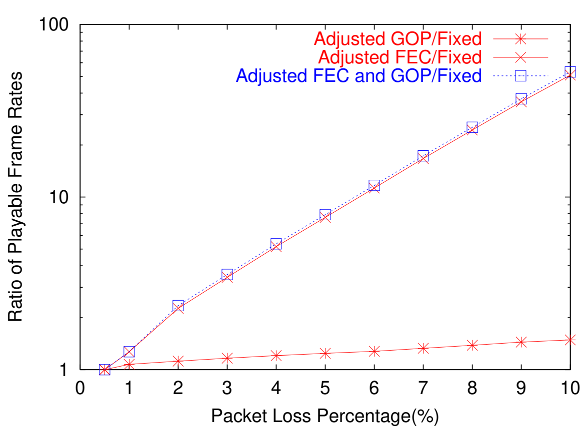
5. ANALYSIS

This section presents results from applying our model from Section 3 with the system settings discussed in Section 4. The goal is to investigate the performance benefits that can be derived from adjustments to GOP and FEC for different network packet loss rates.

5.1 Benefits from Adjusting GOP and Adjusting FEC



a. Playable frame rates



b. Ratio of playable frame rates

Figure 2: Playable frame rates

Figure 2 compares the playable frame rates for the four schemas of Table 3 for a range of packet loss percentages. In Figure 2.b, the three adjustable schemes (adjusted GOP, adjusted FEC, adjusted FEC and GOP) are normalized in terms of the ratio of their optimal playable frame rate versus the playable frame rate in the Fixed schema.

Both graphs in Figure 2 show that adjusting the GOP does not significantly improve the playable frame rate. When the packet loss percentage is higher than 6% the best possible GOP pattern yields an absolute playable frame rate below one frame per second (fps). On the other hand, the benefit from Adjusted FEC is very high. Adjusting FEC with increased packet loss manages to keep the playable frame rate above 4 fps. If the network can keep the packet loss percentage below 5%, adjusted FEC will provide a playable frame rate higher than 10 fps. Above 6% loss, the normalized ratio in Figure 2.b is over 10 for Adjusted FEC. The Adjusted FEC and GOP curves are very close to the FEC curves, suggesting that adjusting GOP patterns (as in [6, 27]) yields insignificant improvement. This suggests that adjusting FEC is the better way to maximize video frame rates in a lossy network than adjusting GOP.

5.2 GOP Behavior

5.2.1 *Adjusting GOP*

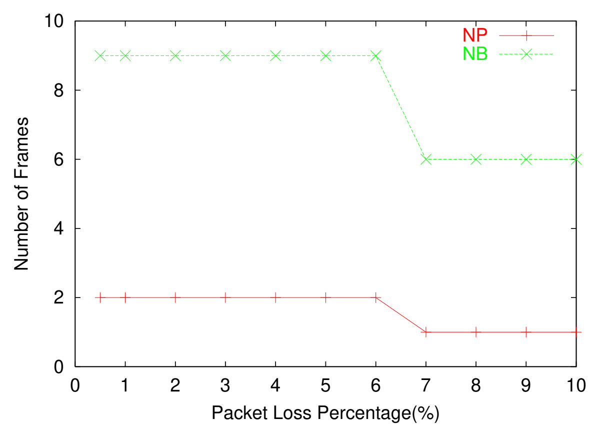


Figure 3: Adjusted GOP

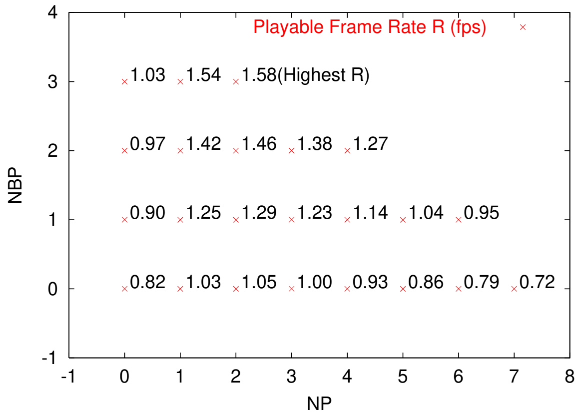
Figure 3 depicts the limited movement in the choice of the optimal GOP pattern in schema B, the Adjusted GOP, as p increases. While N_P does decrease for loss rates above 7%, the optimal N_{BP} is the same (3) for all loss percentages.

In schema D, when both the FEC and GOP spaces are searched, GOP(2,3) ('IBBBPBBBPBBB') is the best choice regardless of the expected packet loss percentage. A change in loss results in only a change in the FEC pattern. Hence, it might be more accurate to label this schema as 'Adjusted FEC, Optimal GOP'.

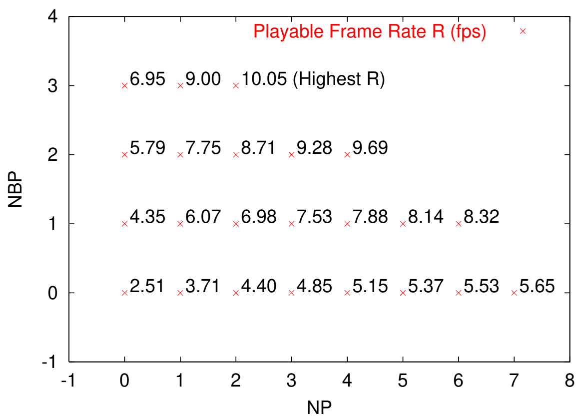
5.2.2 *Fixed GOP*

In the prior discussion the initial Fixed GOP choice of GOP(4,2) was so well-chosen that there was no advantage in trying to adjust GOP to loss expectations. However, this is not necessarily true for every choice of GOP pattern.

To investigate the impact of different GOP choices, we fixed the packet loss percentage at 5% and used our model to compute the playable frame rate R, for possible fixed GOPs. Figure 4.a depicts the playable frame rate R for every



a. Without FEC



b. With Adjusted FEC

Figure 4: Playable frame rate for Fixed GOPs with 5% packet loss

possible GOP when FEC is not used, and Figure 4.b depicts the maximum playable frame rate when FEC is adjusted. Each R in Figure 4 is printed in the two-dimensional grid that corresponds to the x-axis and y-axis values of N_P and N_{BP}, used for the corresponding fixed GOP(N_P,N_{BP}) in the model. The best possible playable frame rate in this two-dimensional space of fixed GOP choices is also indicated with the text 'Highest R' next to the point.

Figures 4a and b indicate that there can be significant performance differences caused by the GOP pattern. For example, with no FEC, GOP(2,3) gets almost twice the playable frame rate as GOP(0,0) (a GOP with only I frames). With Adjusted FEC, the frame rate difference is even larger, with GOP(2,3) getting more than three times the frame rate as GOP(0,0). Thus, it is important to select a good GOP, especially when using FEC.

Knowing these trends, it makes sense to investigate the behavior of FEC by starting with an optimal GOP choice. Although GOP(2,3) is the optimal GOP in Figure 4, we use GOP(4,2) in next section because its performance is very close to that of GOP(2,3) and it has the added advantage of being a very commonly used pattern on the Internet [1].

5.3 FEC Behavior

5.3.1 *Adjusting FEC*

Figure 5 depicts the breakdown of how redundant FEC packets should be adjusted for I, P, and B frames to yield the optimal playable frame rate as the packet loss percentage varies.

In optimally adjusting FEC, the I frames always receive more FEC packets than the P frames. Similarly, the P frames always receive more FEC packets than the B frames. This fits intuition, since this allocation represents both the relative importance and relative sizes of I, P, and B frames respectively.

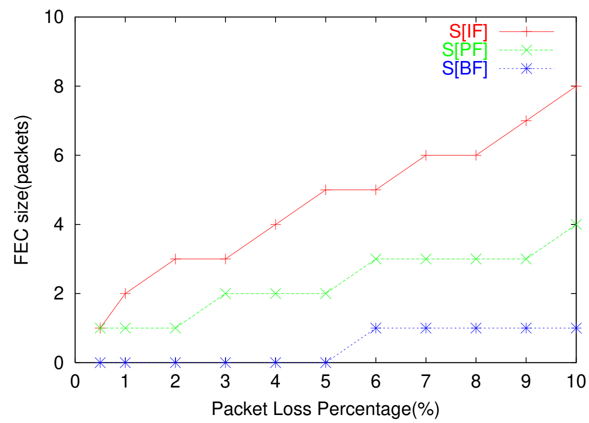


Figure 5: Adjusted FEC pattern

The total number of FEC packets allocated for MPEG increases with the packet loss probability. Increasing the amount of FEC as the loss rate increases is needed to keep the probability of successful frame transmission high. Figure 6 depicts the behavior of the successful frame transmission probability, q, without FEC (Figure 6.a) and with Adjusted FEC (Figure 6.b). Without FEC, Figure 6.a shows that I frames suffer the most as the loss rate increases because they are the largest. However, adjusting FEC as the loss rate increases provides significant protection for all frame types and yields high playable frame rates even in the face of 10% packet loss.

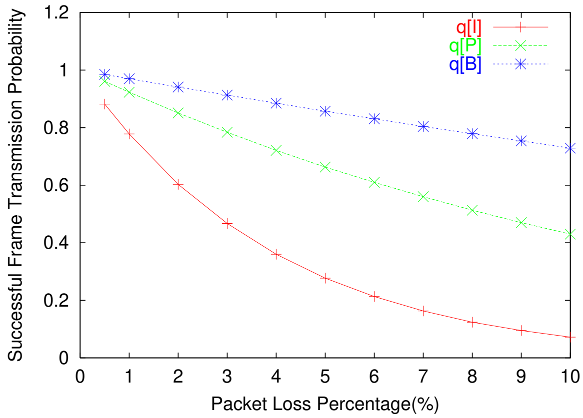
5.3.2 *Fixed FECs vs. Adjusted FEC*

Section 5.2 indicates that adjusting GOP offers little improvement over a fixed GOP when a good GOP pattern is chosen. This section considers whether this still holds when one carefully selects the FEC combination.

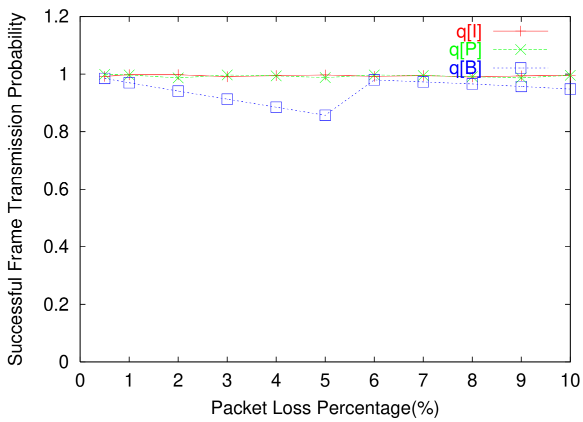
Table 4 shows three fixed FEC choices taken from Figure 5. Small FEC is the adjusted FEC for 0.5% loss, medium FEC is the adjusted FEC for 5% loss, and large FEC is the adjusted FEC for 10% loss.

Fixed FEC	S_{IF}	S_{PF}	S_{BF}
small	1	1	0
medium	4	2	0
large	8	4	1

Table 4: Forward Error Correction Setting

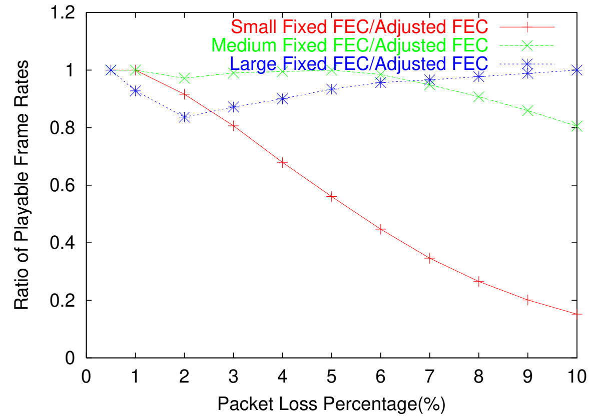


a. Without FEC

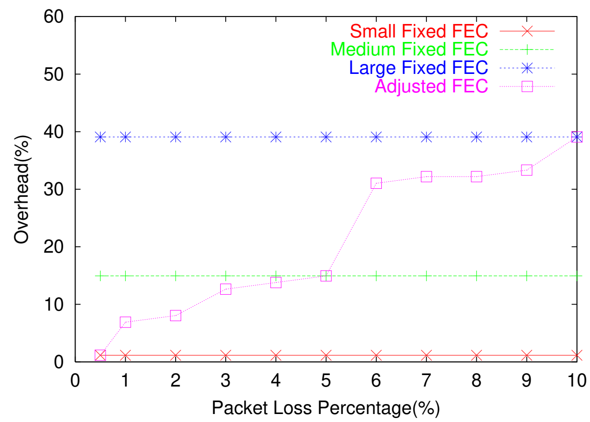


b. With Adjusted FEC

Figure 6: Successful frame transmission probability



a. Ratio of playable frame rates



b. Overhead of Fixed FECs and Adjusted FEC

Figure 7: Fixed FECs vs. Adjusted FEC

Figure 7 compares the playable frame rate of Fixed FEC and Adjusted FEC. The y-axis in Figure 7.a is the ratio between the playable frame rate for a fixed FEC and the playable frame rate for Adjusted FEC. The overhead of the four distinct FEC schemes is plotted in Figure 7.b.

Figure 7 clearly shows adjusting FEC for a given loss probability adds the minimum redundancy needed to achieve the optimum frame rate. The small Fixed FEC adds a low overhead to the MPEG stream and works well with a low packet loss rate, but at higher loss rates, the playable frame rate decreases dramatically since the small Fixed FEC cannot provide enough protection. In contrast, the large Fixed FEC gives adequate protection in the presence of high loss rates, but also introduces a large amount of unnecessary overhead when the loss rates are low.

Also, the total size of all FEC packets is small relative to the I, P, and B frame sizes. For all the experiments from the model that generate the best performance in Figure 5, the FEC overhead is less than 40% of the size of the MPEG frames themselves. This implies that for a moderate cost in FEC redundant packets, significant increases in the playable frame rate can be supported when the network is lossy.

5.4 Effect of Bursty Loss

Our analytic model assumes independent packet loss events, while in practice, losses are often bursty [15, 22]. Bursty losses may reduce the effectiveness of FEC when fewer than K of the N packets in a frame can be recovered, causing a lower playable frame rate.

To evaluate the effects of burty loss, we used a series of traces from an Internet measurement study [7] 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 [15] to provide bursty loss events.

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

Figure 8 depicts the playable frame rates for the simulations along with the playable frame rates estimated by our model. Bursty packet losses do result in the adjusted FEC being less effective, but only marginally. This suggests that assuming packet losses to be independent events in our analytic model to determine FEC does not significantly reduce the model's ability to make very good FEC choices.

5.5 Effect of Variable Round Trip Time

In our analytical model, streaming flow round-trip times (RTTs) are fixed for the entire flow, while in practice, RTTs can vary considerably. The possible impact of variable RTTs

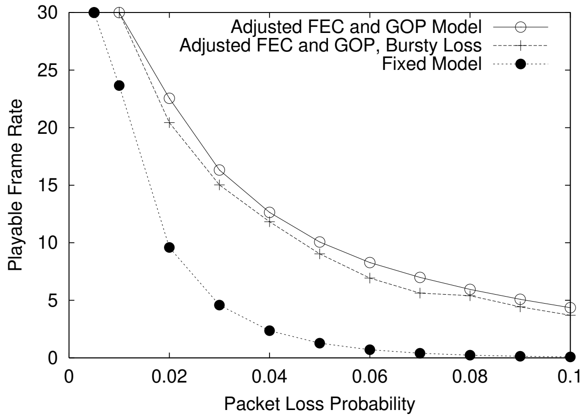


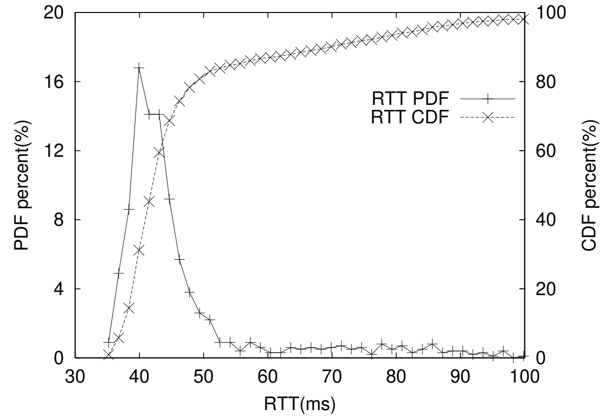
Figure 8: Effect of Bursty Loss

is that the bandwidth estimate by using the fixed average RTT is inaccurate, therefore making the choice of number and distribution of FEC packets less effective.

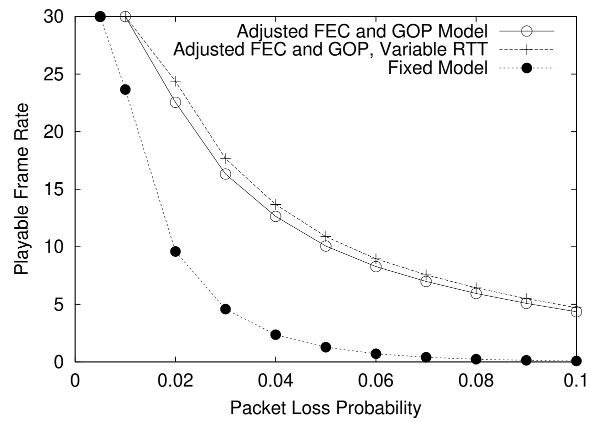
To simulate the effects of variable round-trip times, we selected a trace from [7], depicted in Figure 9a, that had a median RTT of about 50 ms.

We used the analytic model to determine the adjusted FEC and GOP patterns with a fixed RTT of 50 ms. Then, we simulated streaming the MPEG video using the RTT trace and measured the actual frame playout rate at the receiver.

Figure 9b 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 does not come from a normal distribution, 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.



a. RTT Distribution (from [7])



b. Playable Frame Rate

Figure 9: Effect of Variable RTT

6. SUMMARY

This paper presents an analytic model that permits investigation of the impact of adjusting Group of Pictures (GOP) and Forward Error Correction (FEC) on the playable frame rate of TCP-Friendly MPEG video streams. Fully capturing the probabilistic dependencies of MPEG frames, our model determines the optimal way to adjust the distribution of FEC redundancy packets among MPEG frame types and the optimal way to adjust the GOP pattern in the presence of predicted network packet loss. We use the analytic model to search the system state space over a range of network conditions to optimize the playable frame rate for a given loss probability.

Our experiments show that adjusting GOP and FEC provides the best playable frame rate. However, the benefit to playable frame rate from adjusting GOP is slight for typical GOPs, while the benefit to playable frame rate from adjusting FEC is significant, especially for higher loss rates. When an adjusted GOP is used with an adjusted FEC, our model shows a single stable GOP is optimal for all loss percentages. However, the choice of a good GOP is still important as improperly chosen fixed GOPs can degrade the playable frame rate to less than half. Adjusting FEC improves the playable frame rate by adding protection to MPEG frames according to their importance and achieves a higher successful frame transmission probability. The overhead introduced by an adjusted FEC is modest, less than 15% for low loss rates and less than 40% for high loss rates. Moreover, unlike fixed FEC approaches, adjusting FEC under TCP-Friendly constraints does not introduce additional loss or congestion.

There are several rich areas of future work. The model need be evaluated with more realistic network and application characteristics such as the inaccurate loss predication or MPEG with dynamically changing frame sizes. Other possible future research directions include: integrating the results from this paper with a real-time, adaptive streaming MPEG video protocol; extending the analytic model presented here to include other types of media repair, such as media-dependent FEC; and using retransmissions when RTTs are small relative to streaming media playout buffers.

7. REFERENCES

[1] S. Acharya and B. Smith. An Experiment to Characterize Videos Stored on the Web. In *Proceedings of Multimedia Computing and Networking (MMCN)*, Jan. 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] J.-C. Bolot, S. Fosse-Parisis, and D. Towsley. Adaptive FEC-Based Error Control for Internet Telephony. In *Proceedings of IEEE INFOCOM*, Mar. 1999.

[5] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, 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.

[6] J. Chung and M. Claypool. Better-Behaved, Better-Performing Multimedia Networking. In *Proceedings of SCS Euromedia*, May 2000.

[7] 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)*, Apr. 2003.

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

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

[10] 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.

[11] 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.

[12] W. Jiang and H. Schulzrinne. Comparison and Optimization of Packet Loss Repair Methods on VoIP Perceived Quality under Bursty Loss. In *Proceeding of the 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 73–81, May 2002.

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

[14] 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.

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

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

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

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

[19] 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.

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

[26] S.Floyd. Congestion Control Principles. *IETF Request for Comments (RFC) 2914*, Sept. 2000.

[27] J. Walpole, R. Koster, S. Cen, C. Cowan, D. Maier, D. McNamee, C. Pu, D. Steere, and L. Yu. A Player for Adaptive MPEG Video Streaming Over The Internet. In *Proceedings of the SPIE Applied Imagery Pattern Recognition Workshop*, Oct. 1997.