Modeling Best-Effort and FEC Streaming of Scalable Video in Lossy Network Channels

Seong-Ryong Kang, *Student Member, IEEE*, and Dmitri Loguinov, *Member, IEEE*

***Abstract*—Video applications that transport delay-sensitive multimedia over best-effort networks usually require special mechanisms that can overcome packet loss without using retransmission. In response to this demand, forward-error correction (FEC) is often used in streaming applications to protect video and audio data in lossy network paths; however, studies in the literature report conflicting results on the benefits of FEC over best-effort streaming. To address this uncertainty, we start with a baseline case that examines the impact of packet loss on scalable (FGS-like) video in best-effort networks and derive a closed-form expression for the loss penalty imposed on embedded coding schemes under several simple loss models. Through this analysis, we find that the utility (i.e., usefulness to the user) of unprotected video *converges to zero* as streaming rates become high. We then study FEC-protected video streaming, re-derive the same utility metric, and show that for all values of loss rate inclusion of FEC overhead substantially improves the utility of video compared to the best-effort case. We finish the paper by constructing a dynamic controller on the amount of FEC that maximizes the utility of scalable video and show that the resulting system achieves a significantly better PSNR quality than alternative fixed-overhead methods.**

***Index Terms*—FEC rate control, Markov-chain loss, MPEG-4 FGS, utility of video, video streaming.**

I. Introduction

FORWARD-ERROR correction (FEC) is widely used in the Internet for its ability to recover data segments lost in the network [3], [13], [21]. With a proper amount of redundancy included in transmitted packets, FEC can reduce the impact of packet loss on the quality of video, thus improving the performance of streaming over best-effort networks. However, selection of FEC overhead becomes a fairly complicated task when network path dynamics change over time, which in certain cases may lead to reduced or negligible performance gain compared to similar best-effort scenarios [1], [3], [9].

Although FEC appears intuitively beneficial, studies in the literature report conflicting results on its performance in practice. Some of them (e.g., [1], [9]) show that FEC provides little benefit to applications due to the extra overhead, while others (e.g., [5], [7]) find FEC to be promising in the context of particular multimedia applications. To understand the benefits of FEC in Internet streaming, we first analyze the performance of video streaming in best-effort networks and derive a closed-form model for the penalty inflicted on scalable[1] video coding under Markov and renewal patterns of packet loss. For this analysis, we consider *end-user utility* U as the main metric, which we define as the percentage of received data in each frame that can be used for decoding the frame, i.e.,

$$U = \frac{M}{T} \tag{1}$$

where M is the average number of bytes/packets used in decoding a frame and T is the average amount of data per frame successfully delivered to the receiver. Deriving (1) in closed-form, we show that best-effort streaming imposes a significant penalty on video applications when packet loss randomly corrupts the video stream and demonstrate that for any fixed packet loss $p > 0$, the utility $U \to 0$ as the streaming rate goes to infinity.

Given poor performance of best-effort streaming, we next examine FEC-protected transmission of video data. Previous studies in the literature (e.g., [7], [8], [23]) have examined the dynamics of the loss process under a two-state Markov chain and provided numerical models for obtaining the distribution of the number of loss events in a block of fixed size n; however, these models usually rely on complex recursive expressions or tedious summations, neither of which sheds light in *qualitative* or *closed-form* terms on the behavior of FEC in practice. To overcome this limitation and ultimately compute (1), we study the effect of Markov-based packet loss within an FEC block and derive the asymptotic (i.e., assuming large sending rates) distribution of the number of lost packets per FEC block. This model offers a low-complexity version of the same result obtained by the earlier methods and allows computation of other metrics of interest related to FEC streaming.

Armed with this result, we next focus on investigating the performance of video streaming with FEC protection under two-state Markov-chain loss. Assuming that R is the streaming rate of the application and F is the rate of FEC packets, we employ (1) to understand how the FEC overhead rate $\psi = F/(R+F)$, $(0 < \psi < 1)$, affects the utility of received video. Using the models derived in the second part of the paper, we show that U exhibits percolation and converges to 0, 0.5, or $(1-\psi)/(1-p)$ depending on the value of ψ as the streaming rate R approaches infinity. However, for finite R, we find that U achieves a unique

Manuscript received March 11, 2004; revised January 26, 2005, January 27, 2005, September 5, 2005, and January 23, 2006; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor R. Srikant. This work was supported by the National Science Foundation under Grants CCR-0306246, ANI-0312461, CNS-0434940, and CNS-0519442.

The authors are with the Department of Computer Science, Texas A&M University, College Station, TX 77843 USA (e-mail: skang@cs.tamu.edu; dmitri@cs.tamu.edu).

Digital Object Identifier 10.1109/TNET.2006.890110

[1]In scalable video (e.g., MPEG-4 FGS [14]), the enhancement layer is compressed using embedded coding and can be easily re-scaled to match variable network bandwidth during streaming. In such methods, the lower sections of the enhancement layer are more important than the higher sections because their loss renders all dependent data in the source frame virtually useless.

global maximum in some point ψ^* that depends on network packet loss and FEC block size, which indicates *that sending more or less FEC than the optimal amount results in a reduction in* U.

Driven by the goal of maximizing the usefulness of network bandwidth and achieving the highest visual quality under given network conditions, we subsequently explore a simple control mechanism that dynamically adjusts the amount of overhead $\psi(t)$ based on the packet-loss information fed back to application servers by their receivers. We find that such adaptive control allows the application to maintain optimally high utility regardless of the variation in packet loss rates and deliver better PSNR quality to the user compared to schemes with a static or sub-optimal allocation of FEC.

The rest of this paper is organized as follows. Section II discusses related work. Section III studies the impact of packet loss in best-effort networks. Section IV characterizes packet-loss events in an FEC block and Section V analyzes the performance of FEC-based streaming. Section VI describes the proposed mechanism for adjusting the amount of FEC and evaluates its performance in ns2 simulations. Section VII concludes the paper.

II. Related Work

Several studies investigate the performance of FEC; however, the conclusion on its effectiveness generally varies and often depends on rate adjustment mechanisms that are used for including FEC overhead. We next discuss some of the studies in favor of and against FEC.

Altman *et al.* [1] study simple media-specific FEC for audio transmission and show that it provides little improvement to the quality of audio under any amount of FEC. This work uses media-specific FEC that is sometimes less effective in recovering lost packets than media-independent FEC [13]. Biersack *et al.* [3] evaluate the effect of FEC for different traffic scenarios in an ATM network. This study measures the reduction of loss rate for each source and reports that the performance gain of FEC quickly diminishes when all traffic sources employ FEC and the number of sources increases.

Alternative approaches aim to maximize the effect of FEC by choosing the proper amount of overhead and avoiding unlimited rate increase by keeping the combined rate $R + F$ equal to some constant S. Bolot *et al.* [5] present a media-specific method for adjusting FEC overhead under certain constraints on the total sending rate S. That work achieves close to optimal audio-specific subjective quality. Frossard *et al.* [7] propose a method that selects rates R and F using the distortion perceived by end-users. The method is fairly complex since it involves solving recurrence equations, which does not scale to large FEC block sizes.

Note that none of the above studies uses a mechanism that can select the proper amount of overhead dynamically based on network conditions, or offers an explanation of how FEC overhead affects the performance of video applications for a given packet loss rate.

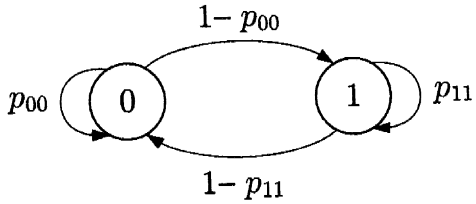


Fig. 1. Two-state Markov chain.

III. Impact of Packet Loss in Best-Effort Networks

In this section, we examine the performance of video streaming in the best-effort Internet assuming random packet loss. We consider two loss models and study the expected amount of recovered data in each video frame. Unlike previous studies (e.g., [2]), we model the dependency between data in each video frame and derive the expected percentage of useful information transmitted over the bottleneck link.

Many studies (e.g., [22]) show that the pattern of Internet packet loss can be captured by Markov models. Thus, we first examine the dynamics of utility (1) assuming that the loss process is a two-state Markov chain. Following the Markov analysis, we study a more general distribution of packet loss and model the network as an alternating ON/OFF process with heavy-tailed ON (loss) and OFF (no loss) periods. While the two modeling approaches are different, they both demonstrate that the best-effort Internet imposes a significant performance penalty on scalable streaming services and its handling of video traffic is far from optimal.

A. Markov Packet Loss

We investigate the effect of packet drops on video quality using the example of MPEG-4 FGS (Fine Granular Scalability) [14].[2] In what follows next, we apply the Markov packet-loss model to FGS sequences, derive the expected amount of useful data recovered from each frame, and define the effectiveness of FGS packet transmission over a lossy channel. Note that in our analysis, we only examine the enhancement layer (which is often responsible for a large fraction of the total rate) and assume that the base layer is fully protected. Even under such conditions, best-effort networks deliver very low performance to scalable flows, which progressively degrades as the streaming rate becomes higher.

Assume that the long-term network packet loss is given by p and the loss process can be modeled by a two-state discrete Markov chain shown in Fig. 1, where states 1 and 0 represent a packet being either lost in the network or delivered to the receiver, respectively. In the figure, $1 - p_{00} > 0$ is the probability that the next packet is lost given that the previous one has arrived and $1 - p_{11} > 0$ is the probability that the next packet

[2]Similar results apply to motion-compensated enhancement layers, which suffer even more degradation under best-effort loss and are not modeled in this work. However, the expected amount of improvement from FEC in such schemes is even higher than that in FGS.

TABLE I
EXPECTED NUMBER OF USEFUL PACKETS (MARKOV MODEL)

Packet loss	$H = 100$		$H = 1,000$	
p	Simulations	Model (3)	Simulations	Model (3)
0.0001	99.595	99.595	960.988	960.986
0.01	68.329	68.324	123.715	123.709
0.1	11.248	11.247	11.255	11.250
0.2	4.998	5.000	4.999	5.000
0.9	0.138	0.138	0.139	0.138

is received given that the previous one has been lost. In the stationary state, probability π_0 and π_1 to find the process in each of its two states are given by:

$$\pi_0 = \frac{1-p_{11}}{2-p_{00}-p_{11}}, \quad \pi_1 = p = \frac{1-p_{00}}{2-p_{00}-p_{11}}. \tag{2}$$

Assume that FGS frame sizes H_j are measured in packets and are given by i.i.d. random variables. The exact distribution of $\{H_j\}$ is not essential and typically depends on the coding scheme, frame rate, variation in scene complexity, and the bitrate of the sequence. The question we address next is *what is the expected amount of useful packets that the receiver can decode from each frame under p-percent random loss?* To answer this question, we denote by Z_j the number of consecutively received packets in a frame j and next compute its expectation $E[Z_j]$, which plays an important role in determining the utility of received video.

Assume that the chain is stationary at the beginning of a frame and let $E\left[Z_j^H\right]$ be the expected number of useful packets per frame if all frames are of size H. Then, we have the following result.

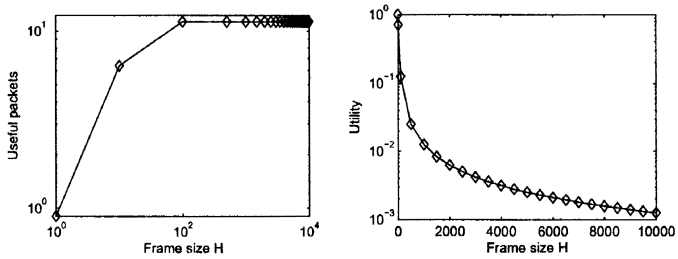
Theorem 1: Assuming a two-state Markov packet loss in (2) and fixed-size frames with $H_j = H$, the expected number of useful packets in each frame is:

$$E\left[Z_j^H\right] = \frac{1-p}{1-p_{00}}\left(1-p_{00}^H\right). \tag{3}$$

Proof: Assume that D_j is the random distance in packets from the beginning of frame j before the first packet-loss event. Let X_1 be the state of the Markov chain when the first packet in frame j passes through the network. Note that if $X_1 = 1$, then the amount of recovered data in the frame is $Z_j = 0$; however, if the loss process is in state $X_1 = 0$, then the recovered amount depends on the value of D_j, i.e., the decoder recovers $Z_j = D_j$ packets when $D_j \le H$ and all $Z_j = H$ packets otherwise. Then, we can write:

$$\begin{aligned} E\left[Z_j^H\right] &= \pi_0 E\left[Z_j^H | X_1 = 0\right] + \pi_1 E\left[Z_j^H | X_1 = 1\right] \\ &= \pi_0 \left(\sum_{i=1}^{H} iP(D_j = i | X_1 = 0) \right. \\ &\quad \left. + H \sum_{i=H+1}^{\infty} P(D_j = i | X_1 = 0) \right). \end{aligned} \tag{4}$$

Conditioning on $X_1 = 0$, it immediately follows that D_j are geometric random variables with a conditional PMF $P(D_j = i | X_1 = 0) = p_{00}^{i-1}(1-p_{00})$, $i \ge 1$. Substituting $\pi_0 = 1-\pi_1 = 1-p$ in (4) and expanding the PMF of D_j, we get (3). ■



(a) $E[Z_j^H]$ (b) U^H

Fig. 2. Simulation results of $E\left[Z_j^H\right]$ and U^H for $p = 0.1$.

To verify (3), we simulate a Markov loss process in Matlab with several values of packet loss p and keep probability p_{00} equal to $1 - 0.8p$ and p_{11} equal to $(2p - 1 + p_{00}(1-p))/p$ so that the average loss rate is p. For this example, we generate a sequence of 10 million frames of size H each and randomly corrupt them using a long Markov-chain loss sequence. Then, we examine each frame j to obtain Z_j and compare $E\left[Z_j^H\right]$ to the model in Table I for $H = 100$ and $H = 1{,}000$. As the table shows, (3) matches simulations very well. Also observe in the table that for $H = 100$ and a reasonably low packet loss of 1%, the expected number of useful packets in each frame is only 68 even though the decoder successfully receives (on average) 99 packets per frame. When we use larger frames with $H = 1{,}000$, the decoder can use only 123 packets on average out of each 990 packets it receives over the network. Moreover, the table shows that under $p = 10\%$, only 11 useful packets are recovered from each frame *regardless of the actual size of the frame*. This means that the bottleneck link under these conditions transmits 8 ($H = 100$) to 90 ($H = 1{,}000$) times more packets than the receiver is able to utilize in decoding its video.

It is easy to notice in (3) that $E\left[Z_j^H\right]$ saturates at $(1-p)/(1-p_{00})$ as $H \to \infty$ (i.e., streaming rates become high). This is shown in Fig. 2(a) for $p = 0.1$ ($p_{00} = 0.92$ and $p_{11} = (2p-1+p_{00}(1-p))/p = 0.28$), in which the number of useful packets recovered per frame indeed converges to $(1-p)/(1-p_{00}) = 11$ as H becomes large.

Rewriting (1) using (3), we have for constant frame sizes:

$$U^H = \frac{E\left[Z_j^H\right]}{H(1-p)} = \frac{1-p_{00}^H}{(1-p_{00})H}. \tag{5}$$

For instance, we get $U^H = 0.12$ for $p = 0.1$ (using the same value of p_{00}, p_{11} as before) and $H = 100$, which means that only 12% of the received FGS packets are useful in enhancing the base layer. The trend of (5) is illustrated in Fig. 2(b), which plots the utility of best-effort streaming for different values of H and $p = 0.1$. As the figure shows, U^H drops to zero inverse proportionally to the value of H, which means that as $H \to \infty$, the decoder receives "junk" data with probability $1 - o(1)$.

Next, we briefly study the result of Theorem 1 for arbitrary frame-size distributions. For this purpose, we expand (3) to variable frame sizes H_j.

Corollary 1: Assuming a two-state Markov packet loss in (2), the expected number of useful packets in each frame is:

$$E[Z_j] = \frac{1-p}{1-p_{00}}\left(1 - E\left[p_{00}^{H_j}\right]\right). \tag{6}$$

TABLE II
EXPECTED NUMBER OF USEFUL PACKETS (VARIABLE FRAME SIZE)

$E[H_j]$	Simulations	Model (6)	Upper bound (3)
10	3.427	3.450	5.861
100	7.446	7.406	8.999
200	8.143	8.148	9.000
500	8.651	8.696	9.000

In the next theorem, we show that $E[Z_j]$ in *any* video sequence with the average frame size H is upper-bounded by (3).

Theorem 2: For a given average frame size $E[H_j] = H$ and Markov-chain loss, the expected number of useful packets per frame is always upper-bounded by that in sequences with $H_j = H$:

$$E[Z_j] \le E\left[Z_j^H\right]. \tag{7}$$

Proof: Set $u(x) = p_{00}^x$ and notice that $u(x)$ is a strictly convex function of x. Then, using Jensen's inequality, it follows that $E[u(H_j)]$ is no less than $u(E[H_j])$ and therefore $1 - E\left[p_{00}^{H_j}\right] \le 1 - p_{00}^H$. Applying this observation to (3) and (6), we immediately obtain (7). ∎

We illustrate the result of Theorem 2 assuming a lognormal[3] frame-size distribution, whose probability density function (PDF) is given by:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma x} e^{-(\log x - \mu)^2/2\sigma^2}, \tag{8}$$

where μ and σ^2 are, respectively, the mean and variance of $\log(H_j)$. For the sake of this example, we use $p = 0.1$, $p_{00} = 1 - p$, $\sigma = 1.5$ and compute μ such that the mean of the lognormal distribution $E[H_j] = e^{\mu + \sigma^2/2}$ matches the desired values. Table II shows the expected number of useful packets in each frame of this sequence and the same metric in the case of constant frame sizes $H_j = H$. As the table shows, $E[Z_j]$ matches simulations well and is in fact upper-bounded by $E\left[Z_j^H\right]$.

Similar observations apply to utility U, which we define as:

$$U = \frac{E[Z_j]}{(1-p)E[H_j]} = \frac{1 - E\left[p_{00}^{H_j}\right]}{(1-p_{00})E[H_j]}. \tag{9}$$

From Theorem 2, it immediately follows that U is upper-bounded by U^H:

$$U \le U^H = \frac{1 - p_{00}^H}{(1-p_{00})H}. \tag{10}$$

This result indicates that regardless of the frame-size distribution, Markov loss implies that $U \to 0$ as $H \to \infty$ and the convergence rate is no worse than linear.

The next question we address is *how many useful packets can be recovered in each frame if the pattern of network packet loss deviates from the Markov model?* We penetrate this problem by obtaining $E\left[Z_j^H\right]$ under a more general packet-loss pattern. Note that since the exact distribution of H_j is application-specific (i.e., unknown) and to conserve space, the rest of the paper only deals with constant frame sizes and no longer considers variable H_j.

[3]Several studies have shown that MPEG frame sizes can be modeled by a lognormal distribution [16], which explains our interest in it.

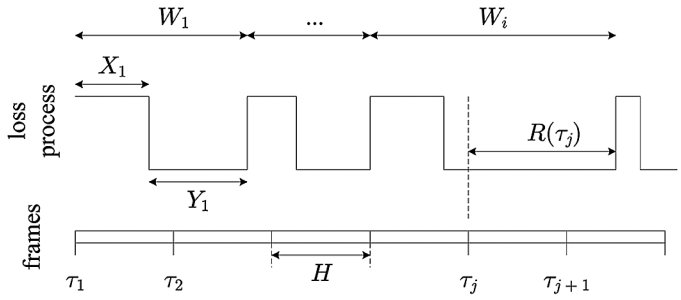


Fig. 3. ON/OFF process $V(t)$ (top) and the transmission pattern of video frames (bottom).

B. Renewal Packet Loss

Several studies have analyzed the characteristics of Internet packet loss and reached a number of conclusions on the distribution of loss-burst lengths including that loss-burst lengths could be modeled as exponential (e.g., [22]) as well as heavy-tailed (e.g., [10]). We overcome this uncertainty by deriving closed-form models for both cases, as well as the more generic case when loss-burst lengths have an arbitrary distribution.

We explore the recurrent behavior of packet loss using a simple stochastic model from renewal theory. Assume that the packet loss process $V(t)$ goes through ON/OFF periods, where all packets are lost during each ON period and all packets are delivered during each OFF period. Then, we can write:

$$V(t) = \begin{cases} 1 & \text{loss at time } t \\ 0 & \text{no loss at time } t \end{cases}. \tag{11}$$

Suppose that the duration of the i-th ON period is given by a random variable X_i and the duration of the i-th OFF period is given by Y_i (X_i and Y_i may be drawn from different distributions). Fig. 3 illustrates the evolution of alternating process $V(t)$. The figure also shows that if $V(t)$ is sampled at a random instant τ_j where frame j starts and the process happens to be in the OFF state, the distance to the next packet loss is given by some residual process $R(\tau_j)$, whose distribution determines Z_j. We elaborate on this observation next.

Assume that X_i and Y_i are independent of each other and sets $\{X_i\}$ and $\{Y_i\}$ consist of i.i.d. random variables. Then, $V(t)$ is an alternating renewal process, whose j-th renewal cycle has duration $W_j = X_j + Y_j$ and whose n-th renewal occurs at time epoch $T_n = \sum_{j=1}^{n} W_j$. Next, notice that long-term network packet loss p is the fraction of time that the process spends in the ON state, which allows us to write:

$$p = \lim_{t\to\infty} P(V(t) = 1) = \frac{E[X_i]}{E[X_i] + E[Y_i]}. \tag{12}$$

Given network packet loss p, we are primarily interested in the location of the first ON event after each frame starts, which determines the number of consecutively received packets in that frame. Suppose that τ_j represents the time instants when the j-th frame starts its transmission over the network. Then, we can safely assume that points τ_j are *uncorrelated* with the cycles

TABLE III
EXPECTED NUMBER OF USEFUL PACKETS (EXPONENTIAL MODEL)

Packet loss	$H = 100$		$H = 1,000$	
p	Simulations	Model (18)	Simulations	Model (18)
0.0001	99.491	99.491	951.599	951.530
0.01	62.582	62.579	98.970	98.995
0.1	8.998	8.999	8.978	9.000
0.2	3.987	4.000	3.979	4.000
0.9	0.111	0.111	0.110	0.111

of packet loss $V(t)$ since the former is an application-specific parameter, while the latter depends on many factors (such as network congestion and cross-traffic) that are not related to the *contents* of the streaming traffic. Thus, we can view τ_j as being uniformly distributed within each renewal cycle of $V(t)$ and $R(\tau_j)$ as the *residual* life of Y_i before the next renewal.

Notice that at τ_j, there are two possible scenarios:

- $V(t)$ is in the ON state;
- $V(t)$ is in the OFF state.

In the former case, the amount of useful data recovered in the frame is $Z_j^H = 0$. However, in the latter case, this amount will depend on the residual life $R(\tau_j)$ of the current OFF cycle (see Fig. 3). Denote by $F_Y(x)$ the distribution of Y_i and assume that $E[Y_i] < \infty$. Next, define $F(x) = P(R(t) \le x)$ to be the distribution of the residual lifetime of the current OFF cycle. Then, recall that $F(x)$ can be expressed as [20]:

$$F(x) = \frac{1}{E[Y_i]} \int_0^x (1 - F_Y(u))\, du. \tag{13}$$

Noticing that the distribution of X_i does not affect $E\left[Z_j^H\right]$, we have the following result.

Theorem 3: Assuming a fixed frame size H, the expected number of useful packets in a frame is determined solely by the distribution of inter-loss durations Y_i and equals:

$$E\left[Z_j^H\right] = (1-p) \int_0^H \bar{F}(x) dx, \tag{14}$$

where $\bar{F}(x) = 1 - F(x)$ is the tail distribution of $R(t)$.

Proof: Consider a frame j that starts at time instant τ_j. Conditioning on $V(t)$ being in the OFF state at τ_j, the number of recovered bits/bytes is the random variable $Z_j^H = \min(R(\tau_j), H)$, which leads to the following:

$$E\left[Z_j^H\right] = (1-p) \left(\int_0^H x f(x) dx + H \int_H^\infty f(x) dx \right), \tag{15}$$

where term $1 - p$ is simply $P(V(t) = 0)$ and $f(x)$ is the PDF of $R(t)$. Using integration by parts, the first integral in (15) becomes:

$$\int_0^H x f(x) dx = HF(H) - \int_0^H F(x) dx. \tag{16}$$

The second integral in (15) is:

$$H \int_H^\infty f(x) dx = H - HF(H). \tag{17}$$

TABLE IV
EXPECTED NUMBER OF USEFUL PACKETS (PARETO MODEL)

Packet loss	$\alpha = 2$, $H = 100$		$\alpha = 3$, $H = 100$	
p	Simulations	Model (19)	Simulations	Model (19)
0.0001	99.492	99.493	99.491	99.492
0.01	68.613	68.621	66.017	66.000
0.1	21.557	21.581	15.012	15.000
0.2	12.062	12.178	7.263	7.272
0.9	0.494	0.501	0.218	0.217
Packet loss	$\alpha = 2$, $H = 1,000$		$\alpha = 3$, $H = 1,000$	
p	Simulations	Model (19)	Simulations	Model (19)
0.0001	952.713	953.006	952.256	952.285
0.01	237.278	237.391	164.950	165.000
0.1	41.652	41.536	17.630	17.647
0.2	21.296	21.213	7.894	7.920
0.9	0.754	0.755	0.225	0.221

Adding (16) and (17) and rearranging the terms, we establish (14). ∎

Note that (13) is based upon the limiting distributions of conventional renewal theory, which provides an *asymptotic* result on $R(t)$ as $t \to \infty$. In order to examine the accuracy of the model for finite $t \ll \infty$, we obtain closed-form expressions of (14) for exponential and Pareto distributions of Y_i in the next two lemmas and compare these results to simulations.

Lemma 1: For exponential Y_i with rate λ, the expected number of useful packets in a frame is

$$E\left[Z_j^H\right] = \frac{1-p}{\lambda} \left(1 - e^{-\lambda H}\right). \tag{18}$$

We illustrate the usage of (18) using exponential Y_i with several values of packet loss p. We set $E[X_i] = 1/(1-p)$, $E[Y_i] = 1/p$ (which leads to $\lambda = p$), and generate over 20 million random values X_i, Y_i to simulate the evolution of ON/OFF process $V(t)$ and obtain metric $E\left[Z_j^H\right]$ for a video stream of fixed-size frames. Model (18) is compared to simulation results for $H = 100$ and $H =$ 1,000 in Table III. As shown in the table, (18) follows simulation results very well and also saturates at fixed values as $H \to \infty$. This result clearly implies that U^H converges toward zero for large H.

The next result shows that heavy-tailed inter-burst gaps Y_i actually improve $E\left[Z_j^H\right]$. In this case, we consider a shifted Pareto distribution $F_Y(x) = 1 - (x/\beta + 1)^{-\alpha}$, where $x \ge 0$, $\alpha > 1$, and $\beta > 0$. Notice that the domain of this distribution is $(0, \infty)$, which allows us to construct a well-formed renewal process and model arbitrarily small durations Y_i.

Lemma 2: For Pareto Y_i with finite mean $E[Y_i] < \infty$, the expected number of useful packets is:

$$E\left[Z_j^H\right] = \begin{cases} \frac{(1-p)\beta}{2-\alpha} \left[\left(\frac{H}{\beta} + 1\right)^{2-\alpha} - 1 \right] & \alpha \ne 2 \\ (1-p)\beta \log\left(\frac{H}{\beta} + 1\right) & \alpha = 2 \end{cases}. \tag{19}$$

We also verify (19) using simulations with Pareto-distributed Y_i and keep $E[X_i] = 1/(1-p)$, which leads to β being equal to $(\alpha - 1)/p$. We compare simulation results with (19) in Table IV for $H = 100$ and $H =$ 1,000. First, notice in the table that simulations match the model very well. Second, observe that

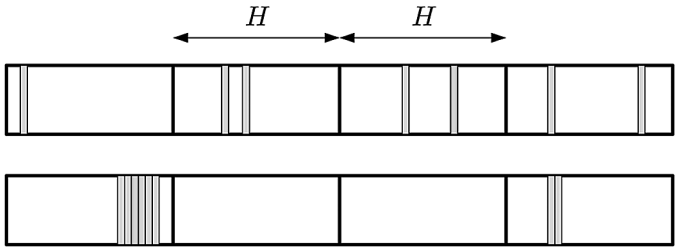


Fig. 4. Packet loss patterns for exponential (top) and Pareto (bottom) Y_i.

the Pareto case delivers more useful packets on average than the exponential case previously shown in Table III. This can be explained by the properties of Pareto $F_Y(x)$, which tends to create large inter-loss gaps followed by many small ones all hitting the same frame. This is schematically shown in Fig. 4 where the Pareto loss events are more bursty and each frame has a higher probability to start within a very large OFF burst.

Also notice that for $\alpha > 2$, $E\left[Z_j^H\right]$ in (19) converges to a constant equal to $(1-p)\beta/(\alpha-2)$ as $H \to \infty$ and utility U^H asymptotically tends to zero as $1/H$. For $\alpha = 2$, the expected number of recovered packets is approximately $(1-p)\beta \log H$, which grows (albeit slowly) to infinity as $H \to \infty$. Nevertheless, even in this case, $U^H \sim \log(H)/H \to 0$ for sufficiently large frame sizes. Finally, for very heavy-tailed cases of $1 < \alpha < 2$, $E\left[Z_j^H\right]$ is proportional to $H^{2-\alpha}$ and utility $U^H \sim H^{1-\alpha}$ still becomes asymptotically negligible as $H \to \infty$.

C. Discussion

Note that for many Internet applications and protocols (such as TCP), it is typically understood that uniform packet loss has benefits over bursty loss. It is interesting, however, that our results imply that for streaming of embedded video signals, bursty packet drops are more desirable than uniformly random. It is further important to note that video-coding methods that use error concealment may exhibit lower performance under bursty loss, in which case the above conclusions would not necessarily hold. In all other cases, subsequent losses within a given frame have no effect on the already-useless frame data and thus lead to better performance of the application as they allow a larger portion of the *remaining* frames to be loss-free.

We next investigate FEC-based streaming as an alternative to retransmission. We study the characteristics of packet drops in an FEC-block in Section IV and discuss the impact of loss on FEC-protected video in Section V.

IV. Impact of Packet Loss on FEC

The distribution of the number of lost packets and the location of the first loss in a block play an important role in understanding the effectiveness of FEC. This section studies these two metrics and offers a model for each assuming large sending rates.

We start by briefly discussing previous work on Markov loss models and pointing out their shortcomings.

A. Background

The original work by Gilbert [8] examines loss events in communication channels under a two-state Markov chain and provides an error model based on recursive formulas that compute the probability of losing a certain number of packets in a block of a given size. Assume a Markov-chain loss model in Fig. 1 and define $P(u,x)$ to be the probability of losing x packets out of u transmitted ones. Next, notice that by conditioning on the last state of the Markov loss process, probability $P(u,x)$ can be written as follows:

$$P(u,x) = P(u,x|0) + P(u,x|1), \tag{20}$$

where $P(u,x|j)$, $j = 0$ or 1, represents the probability of losing x packets out of u transmitted packets given that the loss process is in state j at the end of the block.

Further note that $P(u,x|0)$ and $P(u,x|1)$ can be written as recursive equations [23]:

$$\begin{aligned} P(u,x|0) &= P(u-1,x|0)p_{00} + P(u-1,x|1)(1-p_{11}) \\ P(u,x|1) &= P(u-1,x-1|1)p_{11} \\ &\quad + P(u-1,x-1|0)(1-p_{00}), \end{aligned}$$

where p_{00} and p_{11} represent transition probabilities of the Markov chain shown in Fig. 1.

As two-state Markov loss models have become fairly standard, additional studies (e.g., [7]) examine methods of deriving the above probabilities in closed-form. These approaches and the resulting models are generally very complex both numerically and symbolically. In one example, Yousefizadeh *et al.* [23] recently presented a closed-form solution to the recursive equations above. They derive the probability of receiving $v = u - x$ packets from a u-packet FEC block:

$$\begin{aligned} P(u,x|j) &= p_{00}^{u-2x+j}(1-p_{00})(1-p_{11})^{1-j}W_1P_0 \\ &\quad + p_{00}^{u-2x-1+j}(1-p_{00})^j(1-p_{11})W_2P_1, \end{aligned}$$

where

$$\begin{aligned} W_1 &= \sum_{i=0}^{x-1} \binom{x-1}{i} \binom{v}{i+1-j} \\ &\quad \times (p_{00}p_{11})^{x-1-i}((1-p_{00})(1-p_{11}))^i, \\ W_2 &= \sum_{i=0}^{x-j} \binom{x}{i+j} \binom{v-1}{i} \\ &\quad \times (p_{00}p_{11})^{x-i-j}((1-p_{00})(1-p_{11})^i. \end{aligned}$$

Note that this model holds only for particular conditions (such as $u \geq 2x+1$) and is computationally intensive for non-trivial u even though it does not require solving recursive equations. Also note that none of the previous models provides explicit information about the distribution of the number of lost packets per block or the location of the first loss event.

In Sections IV-B–E, we examine a new (asymptotic) methodology that computes the probability $P(u,x)$ and related metrics of interest in simple closed-form terms.

B. Basic Model

To investigate how the Markov loss process affects each block of FEC, we define $L(n)$ to be the random number of packets lost in a given block of size $n \geq 1$. Notice that $P(u,x)$ discussed

in the previous subsection is simply $P(L(u)=x)$. Next, define Bernoulli random variables:

$$X_i = \begin{cases} 1 & \text{packet } i \text{ is lost} \\ 0 & \text{otherwise} \end{cases}, \tag{21}$$

where $1 \le i \le n$ is the sequence number of the packet within a given block of size n.

Then, the number of lost packets in the block is $L(n) = \sum_{i=1}^{n} X_i$. Note, however, that since loss events X_i in the block are correlated under Markov loss, the distribution of $L(n)$ as $n \to \infty$ does not follow the de Moivre–Laplace theorem for *independent* random variables. Instead, $L(n)$ converges to a Gaussian distribution $N\left(nE[X_i], Var[\sum X_i]\right)$ as long as $\{X_i\}$ are bounded and exhibit exponentially decaying dependence[4] [17]. In such cases, the approximation error is available explicitly and is of the order of $(\log n)^2/\sqrt{n}$. With this result in mind, normality of $L(n)$ is easy to establish and the only remaining pieces of the model are $\mu = E[L(n)]$ and $\sigma^2 = Var[L(n)]$, which we derive next.

C. Model Parameters

Assuming that the Markov chain is in the stationary state at the beginning of the current block, we easily get:

$$\mu = E[L(n)] = E\left[\sum_{i=1}^{n} X_i\right] = np, \tag{22}$$

which is the same result as in the de Moivre–Laplace theorem for sums of independent variables [12]. Notice, however, that the variance of $L(n)$ does not necessarily equal the usual npq, where $q = 1-p$. To understand the next result, define B to be the transition matrix of the two-state Markov chain in (2):

$$B = \begin{bmatrix} p_{00} & 1-p_{00} \\ 1-p_{11} & p_{11} \end{bmatrix}. \tag{23}$$

Furthermore, denote by λ_1 and λ_2 the eigenvalues of the transition matrix B and define a row vector $\mathbf{v} = (0,1)$. Note that since for all stochastic matrices $\lambda_1 = 1$, it is easy to obtain that $\lambda_2 = p_{00} + p_{11} - 1$. Then, we have the following lemma.

Lemma 3: Assume that each loss event in a block of size n follows the two-state Markov chain in (2) and the chain is in the stationary state at time 0. Then, the variance of $L(n)$ is:

$$\sigma^2 = np(1-p) + \frac{2p(1-p)\lambda_2}{1-\lambda_2}\left(n - \frac{1-\lambda_2^n}{1-\lambda_2}\right). \tag{24}$$

Proof: Writing $\sigma^2 = E[L^2(n)] - \mu^2$, we obtain

$$\begin{aligned} E[L^2(n)] &= E\left[\left(\sum_{i=1}^{n} X_i\right)^2\right] \\ &= nE\left[X_i^2\right] + E\left[\sum_{i=1}^{n}\sum_{j\neq i} X_i X_j\right] \\ &= np + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} E[X_i X_j]. \end{aligned} \tag{25}$$

[4]In the context of Markov chains, this means that the chain changes its state with non-zero probability (i.e., $p_{00} \ge \varepsilon, p_{11} \ge \varepsilon$ for some $\varepsilon > 0$) (see Theorem 8.9 in [4]).

Computing $E[X_i X_j]$ in (25), we get

$$E[X_i X_j] = P(X_j = 1 | X_i = 1) P(X_i = 1) = \pi_1 \mathbf{v} B^{j-i} \mathbf{v}^T. \tag{26}$$

Note that the above expression $\mathbf{v}B^{j-i}\mathbf{v}^T$ is simply cell (1,1) of matrix B^{j-i}. Combining (25) and (26), we get:

$$E[L^2(n)] = np + 2p\mathbf{v}\left[\sum_{i=1}^{n-1}\sum_{d=1}^{n-i} B^d\right]\mathbf{v}^T. \tag{27}$$

Next, we compute B^d, which can be represented as a function of B's eigenvalues by Sylvester's theorem [15]. For 2×2 matrices, this leads to a simple closed-form result:

$$B^d = \lambda_1^d \frac{\lambda_2 I - B}{\lambda_2 - \lambda_1} + \lambda_2^d \frac{\lambda_1 I - B}{\lambda_1 - \lambda_2} \tag{28}$$

where I represents the 2×2 identity matrix.

Finally, substituting (28) into (27) and recalling that we only need cell (1,1) in B^d, we get:

$$\begin{aligned} E[L^2(n)] &= np + 2p\sum_{i=1}^{n-1}\sum_{d=1}^{n-i}\left[p + (1-p)\lambda_2^d\right] \\ &= np + 2p\left[p\frac{n(n-1)}{2} + R(n)\right], \end{aligned} \tag{29}$$

where

$$R(n) = \frac{(1-p)\lambda_2}{1-\lambda_2}\left(n - \frac{1-\lambda_2^n}{1-\lambda_2}\right). \tag{30}$$

Combining (22), (29), and (30), we get (24). ■

Simulations confirm that (24) is exact (for examples, see Table V discussed later).

D. Asymptotic Approximation

In this subsection, we examine asymptotic (i.e., for large[5] n) characteristics of the distribution of $L(n)$. Note that unless $2 - p_{00} - p_{11} = 1 - \lambda_2$ is on the order of $1/n$, the term λ_2^n in (24) is virtually zero for non-trivial $n \gg 1$. Since p is fixed and B cannot depend on n, we can drop λ_2^n in (30) to obtain:

$$R(n) \approx \frac{(1-p)\lambda_2}{1-\lambda_2}\left(n - \frac{1}{1-\lambda_2}\right). \tag{31}$$

This leads to the following approximation on $\sigma^2 = np(1-p) + 2pR(n)$ for large n:

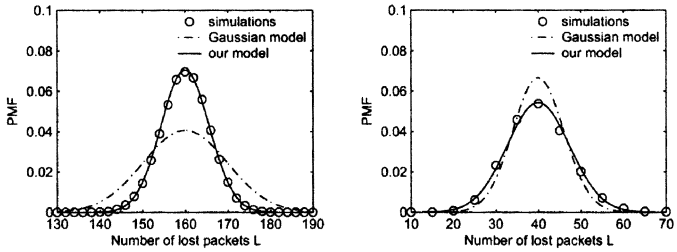
$$\sigma^2 \approx np(1-p) + \frac{2p(1-p)\lambda_2}{1-\lambda_2}\left(n - \frac{1}{1-\lambda_2}\right). \tag{32}$$

Notice that when $R(n)/n \approx 0$, the model above simplifies to the case of Bernoulli loss and σ^2 reduces to $npq(1+o(1))$. Thus, term $R(n)$ determines the amount of "dependency" in the sequence and the amount of deviation of σ^2 from its uncorrelated version. For convenience, re-write (32) as:

$$\sigma^2 = np(1-p)\left(1 + \frac{2\lambda_2}{1-\lambda_2}\left[1 - \frac{1}{(1-\lambda_2)n}\right]\right). \tag{33}$$

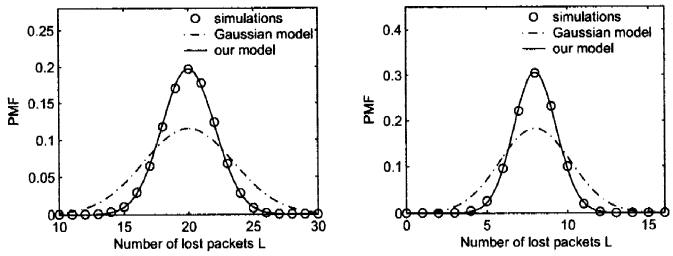
As a result of this transformation, it is easy to observe as a self-check that $\sigma^2 \approx npq$ when $\lambda_2 \approx 0$ (or $p_{00} + p_{11} \approx 1$), or

[5]Throughout the paper, "asymptotically large n" means that $np \gg 0$.



(a) $p = 0.4$ (b) $p = 0.1$

Fig. 5. Distribution of $L(n)$ for $n = 400$ and two different p.



(a) $n = 50$ (b) $n = 20$

Fig. 6. Distribution of $L(n)$ for $p = 0.4$ ($p_{00} = 0.4, p_{11} = 0.1$).

in other words, when there is no dependency between X_i and X_j. This indeed reduces the Markov chain to the Bernoulli case and allows the de Moivre–Laplace theorem to hold.

For large np, we can state that for $2 - p_{00} - p_{11} \geq \varepsilon > 0$:

$$\lim_{n\to\infty} \frac{\sigma^2}{n} = p(1-p)\left(1 + \frac{2\lambda_2}{1-\lambda_2}\right). \tag{34}$$

Combining the above discussion into a single approximation, we obtain the following distribution of $L(n)$.

Corollary 2: Assume that each loss event in a block of size n follows the two-state Markov chain in (2), the chain is in the stationary state at time 0, and $2 - p_{00} - p_{11} \geq \varepsilon > 0$. Then, the distribution of $L(n)$ for large n is:

$$L(n) \sim N\left(np, np(1-p)\frac{p_{00}+p_{11}}{2-p_{00}-p_{11}}\right). \tag{35}$$

Next, we verify model (35) using Matlab simulations. We create a Markov process using two different values of p with two sets of transition probabilities (p_{00}, p_{11}). We use $p_{00} = 0.4$, $p_{11} = 0.1$ to obtain large packet loss $p = 0.4$ and $p_{00} = 0.92$, $p_{11} = 0.28$ for smaller packet loss $p = 0.1$. Simulation results are compared to the model in Fig. 5, where the curve "Gaussian model" is the standard distribution $N(np, npq)$ for independent loss events and curve "our model" represents the distribution predicted by (35). As the figure shows, model (35) matches the simulation very well, while the classical Gaussian model exhibits variance inconsistent with that of the actual distribution. Also notice in the figure that the true distribution of $L(n)$ may have σ^2 both smaller and larger than the corresponding value npq. The first example has $p_{00} + p_{11} = 0.5$, which results in $\sigma^2 \approx npq/3$. The second example has $p_{00} + p_{11} = 1.2$, which leads to $\sigma^2 = 1.5npq$. Further note that (35) holds for relatively small n as well. Fig. 6 shows two examples for $n = 50$ and $n = 20$, respectively, where the match is just as good as in Fig. 5.

Numerical assessment of the model is shown in Table V, which illustrates several examples from the CDF tail of both distributions in Fig. 6. As the table shows, for both values of n, (35) matches simulations very well.

E. Nonstationary Initial State

We now tackle the issue of non-stationary initial distribution of X_0, which is the state of the packet preceding the first packet in the FEC block. This analysis will be required later for the derivation of streaming utility U^H. Define $L_c(n)$ to be the random number of packets lost in a given block of size n conditioned on the initial state X_0 being 1 and μ_c to be its mean:

TABLE V
COMPARISON OF (35) TO SIMULATIONS ($p = 0.4$)

n	Metric	Simulations	Model (35)	Gaussian model
50	μ	20.000	20.000	20.000
	σ^2	4.106	4.106	12.000
	$P(L(n) \leq 16)$	0.043	0.042	0.156
	$P(L(n) \leq 24)$	0.989	0.986	0.903
20	μ	8.000	8.000	8.000
	σ^2	1.706	1.706	4.800
	$P(L(n) \leq 6)$	0.121	0.125	0.246
	$P(L(n) \leq 10)$	0.979	0.972	0.873

$$\mu_c = E[L_c(n)] = E[L(n)|X_0 = 1]. \tag{36}$$

Lemma 4: Assume that each loss event in a block of length n follows the two-state Markov chain in (2). Then, the mean of $L_c(n)$ for large n is:

$$\mu_c = np + (1-p)\lambda_2 \frac{1-\lambda_2^n}{1-\lambda_2}. \tag{37}$$

Proof: Note that the mean of $L(n)$ conditioned on the value of initial state $X_0 = x$ is:

$$E[L(n)|X_0 = x] = \sum_{i=1}^{n} P(X_i = 1|X_0 = x). \tag{38}$$

To obtain $P(X_i = 1|X_0 = x)$, we need cell $(x, 1)$ from the matrix B^i. From (28), we easily establish that:

$$P(X_i = 1|X_0 = x) = \begin{cases} \pi_1\left(1-\lambda_2^i\right), & x = 0 \\ \pi_1 + (1-\pi_1)\lambda_2^i, & x = 1 \end{cases}. \tag{39}$$

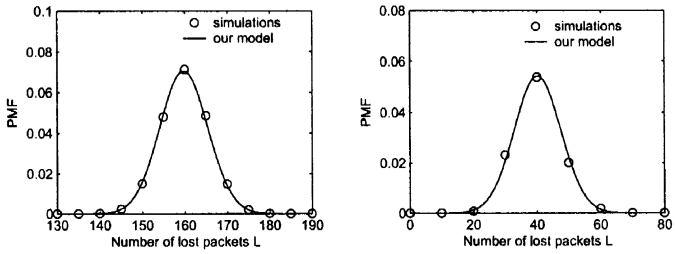
Setting $x = 1$ and expanding (38) using (39), we get (37). ■

Simulations confirm that (37) is exact. For large n, the term λ_2^n becomes negligible and thus (37) can be simplified to:

$$\mu_c \approx np + \frac{\lambda_2(1-p)}{1-\lambda_2}. \tag{40}$$

Next, define σ_c^2 to be the variance of $L_c(n)$:

$$\sigma_c^2 = Var[L_c(n)] = Var[L(n)|X_0 = 1]. \tag{41}$$



(a) $p = 0.4$ (b) $p = 0.1$

Fig. 7. Distribution of $L_c(n)$ for $n = 400$ and two different p.

Then, we have the following result.

Lemma 5: Assume that each loss event in a block of length n follows the two-state Markov chain in (2). Then, the variance of $L_c(n)$ for large n is:

$$\sigma_c^2 \approx (p-1)\left[np + p + \frac{1-2np-6p}{\gamma} + \frac{5p-1}{\gamma^2}\right], \quad (42)$$

where $\gamma = 2 - p_{00} - p_{11} \geq \varepsilon > 0$.

Proof: Write the conditional variance σ_c^2 as:

$$\sigma_c^2 = E\left[L^2(n)|X_0 = 1\right] - \mu_c^2. \quad (43)$$

Then, we can express the first term of (43) as:

$$E\left[L^2(n)|X_0 = 1\right] = E[L(n)|X_0 = 1] + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} E[X_i X_j|X_0 = 1]. \quad (44)$$

Notice that by conditioning on $X_0 = x$, $E[X_i X_j]$ depends on the value of i in addition to the distance $d = j - i$:

$$E[X_i X_j|X_0 = x] = \mathbf{v}B^d\mathbf{v}^T P(X_i = 1|X_0 = x). \quad (45)$$

Using (45) and (39), we obtain:

$$E[X_i X_j|X_0 = 1] = \left(p + (1-p)\lambda_2^{j-i}\right)\left(p + (1-p)\lambda_2^i\right). \quad (46)$$

Denoting by G the double summation term in (44) and expanding it using (46), we have:

$$\begin{aligned} G &= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} E[X_i X_j|X_0 = 1] \\ &= \sum_{j=1}^{n-1}\left(p^2 j - p + p^2 + p(1-p)\frac{1-\lambda_2^{j+1}}{1-\lambda_2}\right) \\ &\quad + \sum_{j=1}^{n-1}\left((p^2-p)j\lambda_2^j - \frac{(1-p)^2}{1-\lambda_2}\lambda_2^{n+1}\right) \\ &\quad + \sum_{j=1}^{n-1}\left(np(1-p) - (1-p)^2 + \frac{(1-p)^2}{1-\lambda_2}\right)\lambda_2^j. \quad (47) \end{aligned}$$

TABLE VI
COMPARISON OF (49) TO SIMULATIONS ($p = 0.4$)

n	Metric	Simulations	Model (49)
400	μ_c	159.800	159.800
	σ_c^2	32.053	32.053
	$P(L_c(n) \leq 142)$	0.001	0.001
	$P(L_c(n) \leq 150)$	0.051	0.050
	$P(L_c(n) \leq 170)$	0.971	0.970

Let G_1, G_2, and G_3 be the first, second, and third summations in (47), respectively. Expanding each term separately, we get:

$$\begin{aligned} G_1 &= (n-1)\left(\frac{p^2 n}{2} - p + p^2 + \frac{p(1-p)}{1-\lambda_2}\right) \\ &\quad - \frac{p(1-p)\lambda_2^2\left(1-\lambda_2^{n-1}\right)}{(1-\lambda_2)^2}, \\ G_2 &= (p^2-p)\lambda_2\frac{(1-n\lambda_2^{n-1})(1-\lambda_2) + \lambda_2\left(1-\lambda_2^{n-1}\right)}{(1-\lambda_2)^2} \\ &\quad - \frac{(1-p)^2}{1-\lambda_2}(n-1)\lambda_2^{n+1}, \\ G_3 &= \left((1-p)(np-1+p) + \frac{(1-p)^2}{1-\lambda_2}\right)\frac{\lambda_2\left(1-\lambda_2^{n-1}\right)}{1-\lambda_2}. \end{aligned}$$

Using the same argument for large n as in the previous subsection and dropping terms λ_2^{n-1} and λ_2^{n+1}, we can simplify G_1, G_2, and G_3 to:

$$\begin{aligned} G_1 &= (n-1)\left(\frac{p^2 n}{2} - p + p^2 + \frac{p(1-p)}{1-\lambda_2}\right) - \frac{p(1-p)\lambda_2^2}{(1-\lambda_2)^2}, \\ G_2 &= \frac{(p^2-p)\lambda_2}{1-\lambda_2}\left(1 + \frac{\lambda_2}{1-\lambda_2}\right), \\ G_3 &= \left((1-p)(np-1+p) + \frac{(1-p)^2}{1-\lambda_2}\right)\frac{\lambda_2}{1-\lambda_2}. \quad (48) \end{aligned}$$

Substituting $\lambda_2 = p_{00} + p_{11} - 1$ and using (40) and (48), we obtain (42). ■

Combining (37) and (42), the next asymptotic result follows immediately.

Corollary 3: Assume that each loss event in a block of length n follows the two-state Markov chain in (2). Then, the distribution of $L_c(n)$ for large n is:

$$L_c(n) \sim N\left(\mu_c, \sigma_c^2\right). \quad (49)$$

We next present simulations that show the accuracy of (49). For this example, we use $n = 400$ and two different values of p and plot the distribution of $L_c(n)$ in Fig. 7. To demonstrate the numerical match, we compute several metrics of interest for $p = 0.4$ and compare them with (49) in Table VI. As the figure and table show, (49) agrees with simulations very well.

V. PERFORMANCE OF FEC IN SCALABLE STREAMING

Our next step is to study the performance of FEC-based video streaming considering two loss patterns and analyze the convergence point of U^H as $H \to \infty$.

Since our main interest in FEC is how its overhead affects the utility of received video, we examine a generic media-independent FEC scheme based on (n, k) block codes (such as parity or Reed–Solomon codes), where n is the total number of packets in an FEC block and k is the number of redundant FEC packets in the block. Thus, the actual number of video data packets in each block is $H = n - k$ and the FEC overhead rate (i.e., fraction of FEC packets) ψ is k/n. Recall that under (n, k) block coding, all H data packets are recovered if the number of lost packets in a block is no more than the number of FEC packets k. However, if the channel loses more than k packets, then only those packets in the enhancement layer located *before the first loss in the block* can be used in decoding.

A. Markov Packet Loss

In this subsection, we first investigate the expected amount of data recovered in each block and in Section V-B analyze the corresponding utility of received video.

To derive $E\left[Z_j^H\right]$, we again assume that $L(n)$ is the number of packets lost in a block of size n and define $\bar{Q} = E\left[Z_j^H \mid L(n) > k\right]$ to be the expected number of useful video packets recovered from an FEC block when $L(n)$ is greater than the number of FEC packets in the block. The following result states the value of $\bar{Q}$.

Lemma 6: Assuming a two-state Markov packet loss in (2) and $L(n) > k$, the expected number of useful video packets recovered per frame is:

$$\begin{aligned}\bar{Q} &= E\left[Z_j^H \mid L(n) > k\right] \\ &= \frac{p_{00}p_{11} - \lambda_2}{1 - \lambda_2} \sum_{i=1}^{n-k-1} i p_{00}^{i-1} \frac{P(L_c(n-i-1) > k-1)}{P(L(n) > k)}.\end{aligned} \tag{50}$$

Proof: Assume that D_j is the random distance in packets to the first loss in a block j as before. Then, we can obtain $P(D_j = i)$ using the basic properties of Markov chains:

$$P(D_j = i) = \begin{cases} \pi_1 & i = 0 \\ \pi_0 p_{00}^{i-1}(1 - p_{00}) & i \geq 1 \end{cases}. \tag{51}$$

Next, write $\bar{Q}$ as:

$$\bar{Q} = E\left[Z_j^H | L(n) > k\right] = \sum_{i=1}^{n-k-1} i P(D_j = i | L(n) > k).$$

Using Bayes' formula, we can get:

$$P(D_j = i | L(n) > k) = \frac{P(L(n) > k | D_j = i) P(D_j = i)}{P(L(n) > k)}. \tag{52}$$

Next, note that we can compute:

$$P(L(n) > k | D_j = i) = P(L_c(n - i - 1) > k - 1), \tag{53}$$

which represents the probability of losing more than $k - 1$ packets from $n - i - 1$ transmitted ones conditioned on the $(i + 1)$-st packet being lost.

Finally, recalling that $\pi_0 = (1 - p_{11})/(2 - p_{00} - p_{11}) = (1 - p_{11})/(1 - \lambda_2)$ and with the help of (52), (53), and (51), we get (50). ∎

TABLE VII
COMPARISON OF (50) TO SIMULATIONS

$p = 0.4$, $n = 400$, $\sigma^2 = 32.106$			
k	$P(L(n) > k)$	$\bar{Q}$ in simulations	$\bar{Q}$ in (50)
170	0.0307	0.802	0.807
165	0.1656	0.863	0.865
160	0.4674	0.919	0.920
155	0.7874	0.964	0.963
140	0.9996	0.999	0.999
0	1.0000	1.000	1.000
$p = 0.1$, $n = 400$, $\sigma^2 = 54.943$			
k	$P(L(n) > k)$	$\bar{Q}$ in simulations	$\bar{Q}$ in (50)
50	0.0809	8.245	8.002
45	0.2224	8.905	8.800
40	0.4593	9.604	9.601
30	0.9068	10.813	10.840
20	0.9983	11.231	11.224
0	1.0000	11.250	11.250

Notice that by utilizing the models derived in Section IV, we can compute for asymptotically large n each of the terms in (50) individually, which in turn allows us to calculate $\bar{Q}$. To verify (50), we compute $\bar{Q}$ in simulations and show the result in Table VII. For the first case, we use large packet loss $p = 0.4$ ($p_{00} = 0.4$, $p_{11} = 0.1$), $n = 400$, and over 1 billion iterations. As the table shows, (50) matches simulation results very well. For the second case, we use smaller packet loss $p = 0.1$ ($p_{00} = 0.92$, $p_{11} = 0.28$) and observe in the table that (50) is reasonably accurate as well. It is worth noting that the model is more accurate when np is large or $\sigma^2 < np$. Thus, due to the small np and $\sigma^2 > np$, the match in the second case in Table VII is not as good as that in the first case.

Using the result in (50), we easily get $E\left[Z_j^H\right]$.

Corollary 4: Assuming two-state Markov packet loss with average loss probability p, the expected number of useful packets recovered per FEC block of size n is:

$$E\left[Z_j^H\right] = P(L(n) \leq k)H + P(L(n) > k)\bar{Q}. \tag{54}$$

B. Utility

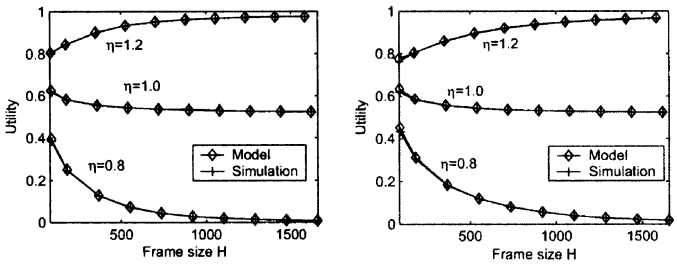
Defining a new metric $C(n) = \bar{Q}P(L(n) > k)$ and re-writing (1) using (54), we get:

$$U^H = \frac{P(L(n) \leq k)H + C(n)}{n(1 - p)}. \tag{55}$$

For convenience of presentation, define the overhead rate ψ as a linear function of packet loss: $\psi = \eta p$ (where η is a constant). Then, we have in the next theorem the asymptotic behavior of U^H as the video rate becomes large.

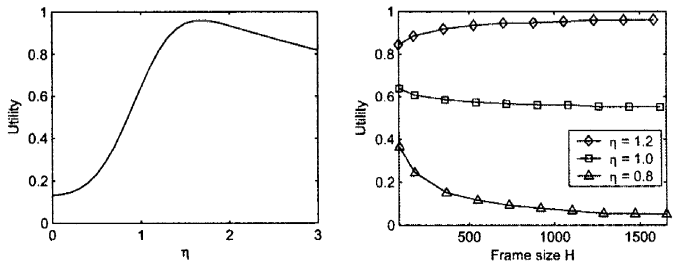
Theorem 4: Assuming a two-state Markov packet loss in an FEC block of size n, average loss probability p, and FEC overhead rate $\psi = \eta p$, $(0 < \psi < 1)$, the utility of received video for each FEC block converges to the following:

$$\lim_{H \to \infty} U^H = \begin{cases} 0 & 0 < \eta < 1 \\ 0.5 & \eta = 1 \\ \frac{1-\psi}{1-p} & 1 < \eta < 1/p \end{cases}. \tag{56}$$



(a) Bernoulli loss (b) Markov loss

Fig. 8. Simulation results of U^H and their comparison to model (56) for Bernoulli loss and Markov loss ($p_{00} = 0.92$, $p_{11} = 0.28$). In both figures, $p = 0.1$.



(a) U^H for finite n (b) Renewal loss

Fig. 9. (a) U^H computed from (55) for $n = 100$ and different values of η. (b) Simulation results of U^H for renewal loss. In both figures, $p = 0.1$.

Proof: Recalling that the distribution of $L(n)$ is asymptotically normal with parameters μ and σ^2 as discussed in Section IV-D, we can write:

$$P(L(n) \leq k) = \int_{-\infty}^{k} f(x)dx, \tag{57}$$

where $f(x)$ is the PDF of $L(n)$.

Define $\phi(x)$ to be the PDF of the standard normal distribution and let $z = (k - \mu)/\sigma$. Then, re-write (57) as:

$$P(L(n) \leq k) = \int_{-\infty}^{z} \phi(x)dx. \tag{58}$$

Using $\psi = \eta p$ and $k = \psi n$, re-write z as:

$$z = \frac{(\eta - 1)\sqrt{(H + k)p}}{\sqrt{(1 - p)\frac{p_{00} + p_{11}}{2 - p_{00} - p_{11}}}}. \tag{59}$$

Notice that $(p_{00} + p_{11})/(2 - p_{00} - p_{11}) > 0$ and observe from (59) that as $H \to \infty$, $z \to -\infty$ if $\eta < 1$, $z \to \infty$ if $\eta > 1$, and $z = 0$ if $\eta = 1$. Thus, the probability $P(L(n) \leq k)$ in (58) converges to the following as $H \to \infty$:

$$\lim_{H \to \infty} P(L(n) \leq k) = \begin{cases} 0 & 0 < \eta < 1 \\ 0.5 & \eta = 1 \\ 1 & 1 < \eta < 1/p \end{cases}. \tag{60}$$

Next, observe in (50) that since $P(L_c(n - i - 1) > k - 1)$ is less than or equal to 1, $C(n)$ is upper-bounded by:

$$C(n) \leq \frac{p_{00}p_{11} - \lambda_2}{1 - \lambda_2} \sum_{i=1}^{n-k-1} i p_{00}^{i-1} = \frac{p_{00}p_{11} - \lambda_2}{1 - \lambda_2} \bar{C}, \tag{61}$$

where

$$\bar{C} = \sum_{i=1}^{n-k-1} i p_{00}^{i-1}. \tag{62}$$

Recalling that $p_{00} > 0$ and expanding (62), we get:

$$\bar{C} = \frac{1 + (n - k - 1)p_{00}^{n-k} - (n - k)p_{00}^{n-k-1}}{(1 - p_{00})^2}. \tag{63}$$

Since $\bar{C}/n \to 0$ as $n \to \infty$, so does $C(n)/n$. Thus, using (60) in (55), and utilizing the fact that $H = n(1 - \psi)$, we immediately get (56). ■

We next verify the asymptotic characteristics of the achieved utility in (56). Before considering a general Markov loss model, we first examine a special case with $p_{00} = 1 - p$ (i.e., Bernoulli loss). Fig. 8(a) plots simulation results of U^H for different η and compare them with (56). As the figure shows, (56) matches simulations very well and U^H indeed converges to 0, 0.5, or $(1 - \psi)/(1 - p) = 0.9778$ as the streaming rate becomes high. For the general Markov loss case, we plot simulation results U^H and compare them with the values predicted by (56) for three different values of η in Fig. 8(b). As the figure shows, U^H follows a trend similar to that in the Bernoulli case with the exception of a slightly slower convergence rate (Markov chains with dependency between the states are more slowly mixing than the Bernoulli case). For instance, under the Markov-chain loss, $U^{1800} = 0.016$ for $\eta = 0.8$, while the Bernoulli case has $U^{1800} = 0.007$ for the same value of η (see Fig. 8).

In summary, the above result on U^H implies that 1) the amount of overhead used in FEC has a significant impact on the quality of received video; 2) U^H asymptotically achieves its maximum when the amount of overhead $\psi = \eta p$ is just slightly larger than the average network loss p. Note, however, that when the streaming rate is finite (i.e., $n < \infty$), U^H depends on n as well as η and the optimal amount of overhead can be determined by minimizing (55). To demonstrate this, we show one such example with finite $n = 100$ and $p = 0.1$ in Fig. 9(a). As the figure shows, U^H reaches its maximum at $\eta = 1.7$, which is much larger than that predicted by (56). We leverage this result later in the paper and next focus on more generic patterns of packet loss.

C. Renewal Packet Loss

In this section, we study U^H under ON/OFF renewal packet loss. Similarly to the result in (54) discussed in Section V-A, we model the amount of useful data recovered from an FEC block as:

$$E\left[Z_j^H\right] = E\left[Z_j^H | L(t) \leq k\right] P(L(t) \leq k) + E\left[Z_j^H | L(t) > k\right] P(L(t) > k), \tag{64}$$

where $L(t) = \int_t^{t+n} V(u)du$ is the number of lost packets in a block of size n starting at time instant t and $V(u)$ is the ON/OFF process described in Section III-B. Unfortunately, computing distribution $P(L(t) \leq k)$ under an ON/OFF renewal process appears to be impossible in closed form even though many studies (e.g., [18]) have attempted this task in the last 50 years. Hence,

TABLE VIII
UTILITIES IN SIMULATION (RENEWAL LOSS)

Packet loss p	Convergence point	$(1-\psi)/(1-p)$
0.01	0.9974	0.9979
0.1	0.9775	0.9777
0.2	0.9496	0.9500
0.4	0.8665	0.8666

we do not pursue this direction further and show instead convergence of U^H using simulations without offering a closed-form model.

For this case, we generate 20 million random values for ON and OFF durations, where each of X_i and Y_i are *i.i.d.* Pareto. We use $E[Y_i] = 10$ and $E[X_i] = E[Y_i]p/(1-p)$ so as to keep the average loss equal to p and plot the simulation results of U^H for different values of η and $p = 0.1$ in Fig. 9(b). As the figure shows, U^H again exhibits a percolation point around $\eta = 1$ (i.e., $\psi = p$) and converges to three different values depending on η.

The final question we address is whether U^H converges to the same values as in the Markov case. We conduct simulations using very large n and several values of p. For each value of p, we identify a convergence point, at which increasing H virtually does not change the value of U^H (change in U^H after doubling the value of H is less than 0.001) and illustrate in Table VIII convergence values of U^H for $\eta = 1.2$. As the table shows, U^H approaches to $(1-\psi)/(1-p)$ regardless of the value of p. This is the same asymptotic result observed in the Markov loss case discussed in the previous subsection. We also found that U^H converges to 0.5 for $\eta = 1$ and 0 for $\eta < 1$, but omit these results for brevity. This demonstrates that as long as the application can measure p, the behavior of U^H for large n is almost the same under many fairly general conditions of network loss.

The next question we address is how to select the proper amount of overhead such that U^H is maximized for a given streaming rate H and network packet loss p.

VI. ADAPTIVE FEC CONTROL

A. Framework

In a practical network environment (such as the Internet), packet loss is not constant and changes dynamically depending on cross traffic, link quality, routing updates, etc. Hence, streaming servers must often adjust the amount of FEC overhead according to changing packet loss to maintain high end-user utility.

To remain friendly to other applications in the Internet and avoid filling network paths with unnecessary FEC packets, a streaming server must comply with the sending rate S suggested by its congestion control algorithm. Given S, the streaming server then determines FEC rate F and video source rate R such that $S = R + F$. Recall that to achieve high end-user utility, overhead rate ψ must be slightly higher than packet loss p as discussed in Section V; however, the exact value of optimal ψ^* depends on the streaming rate and current packet loss p (the latter of which is generally coupled with congestion control and should be provided by its feedback loop).

Next, we discuss a simple approach that can select the proper amount of FEC overhead using our previous analysis. The main problem is how to select optimal η^* for a given packet loss p and FEC block size n to achieve maximum utility. One simple solution is to construct an optimization problem around (55):

$$\eta^* = \arg\max_{\eta} U^H(\eta, p, n), \tag{65}$$

which can be easily solved using binary search and applying models developed in the previous section as long as packet loss p, block size n, and Markov properties of the loss process are known. In practice, this can be implemented by fitting a Markov model to the measured loss events and maximizing utility in (65) regardless of whether the actual network loss exhibits Markovian properties or not. Simulations below suggest that the actual distribution of loss-burst lengths does not have a significant effect on the result.

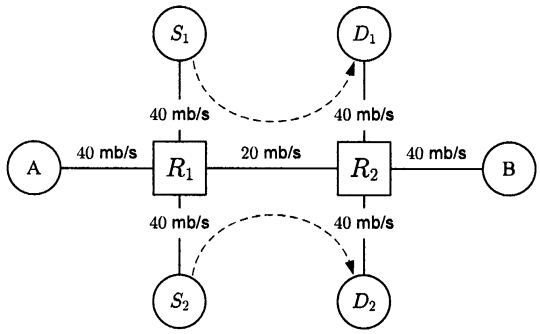


Fig. 10. ns2 simulation topology.

B. Evaluation Setup

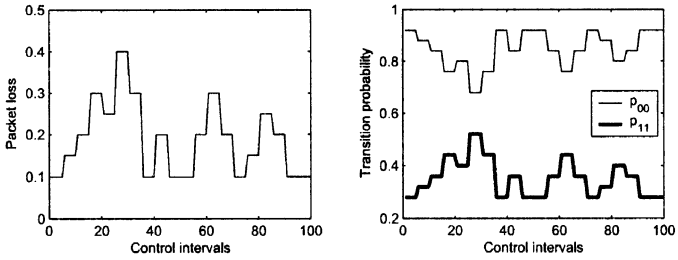
In this section, we present simulation results of our adaptive FEC-based scheme including the properties of U^H and video quality. We first simulate a Markov loss process, obtain packet loss statistics for each video frame, and examine the resulting utility and PSNR video quality of our method in comparison to two approaches that use fixed amounts of FEC overhead. We then conduct ns2 [11] simulations to briefly investigate whether the results obtained from the Markov model are valid in more realistic network environments. In all simulations, one video frame (40,000 bytes without including the base layer) consists of 200 packets, 200 bytes each (these numbers are derived from MPEG-4 coded CIF Foreman with a 128-kb/s base layer coded at 10 frames per second). For convenience of PSNR computation and to keep overhead reasonable, we use FEC block size $n = 200$ packets.

For ns2 simulations, we use a simple topology shown in Fig. 10, in which video source A sends packets at 3.2 mb/s to receiver B over a single bottleneck link of capacity 20 mb/s. To congest the bottleneck link, we use N FTP connections between nodes S_1 and D_1 and 400 HTTP sessions between nodes S_2 and D_2. All access links are 40 mb/s. Each cross-traffic flow starts randomly and N varies over time to produce different values of network packet loss $p(t)$.

We start our investigation with the behavior of utility U^H.

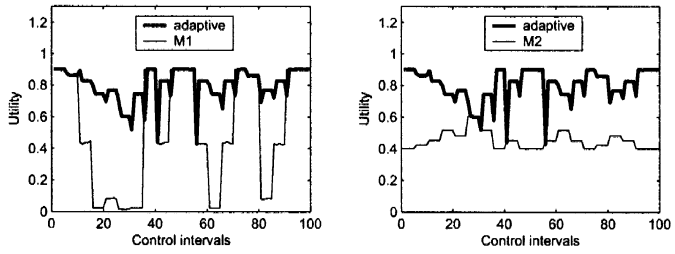
C. Properties of U^H

To illustrate the adaptivity of (65), we first present results based on Markov loss simulations in Matlab. In this example,



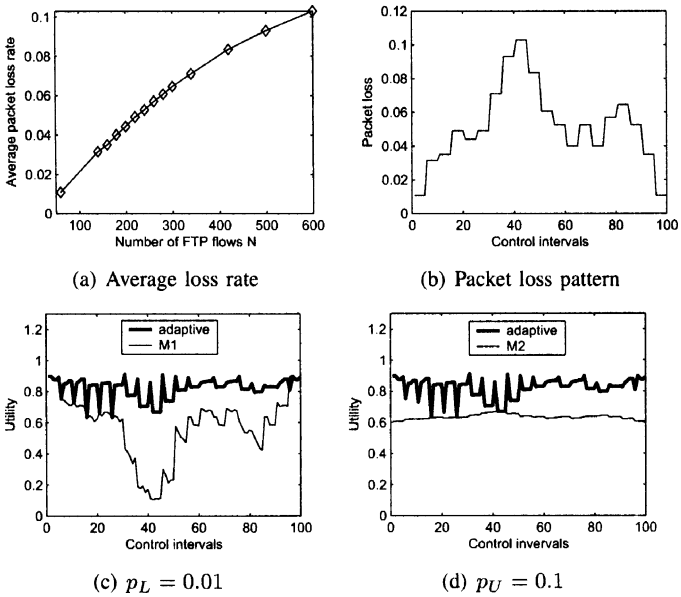
(a) Packet loss pattern (b) p_{00}, p_{11}

Fig. 11. Packet loss pattern obtained through Markov-chain simulation using transition probabilities p_{00} and p_{11}.



(a) $p_L = 0.1$ (b) $p_U = 0.4$

Fig. 12. Metric U^H achieved by the adaptive FEC overhead controller (65) and its comparison to utilities obtained in two different scenarios that use fixed amounts of overhead.



(a) Average loss rate (b) Packet loss pattern

(c) $p_L = 0.01$ (d) $p_U = 0.1$

Fig. 13. (a) Average packet loss rate for different number of FTP flows N in ns2 simulation. (b) Packet loss pattern obtained through ns2 simulation. (c)-(d) Evolution of U^H achieved by the adaptive FEC overhead controller (65) and its comparison to that of utilities obtained in two different scenarios that use fixed amounts of overhead.

we simulate a streaming session with a hypothetical packet loss pattern shown in Fig. 11(a). The evolution of $p(t)$ in Fig. 11(a) is obtained using the Markov chain in (2) with transition probabilities p_{00} and p_{11} plotted in Fig. 11(b). We consider two different fixed-overhead schemes (we call them M_1 and M_2 hereafter) to compare with our adaptive method. To determine the fixed amount of overhead, M_1 and M_2 use the lower ($p_L = 0.1$) and upper ($p_U = 0.4$) bounds on packet loss in Fig. 11(a), respectively.

We plot the achieved utility of FEC-protected video in Fig. 12. As the figure shows, (65) maintains its utility very high (in fact approaching the optimal value of U^H) along the entire streaming session with small deviations only at points when $p(t)$ transitions to its new value. Also observe in the figure that fixed-overhead schemes M_1 and M_2 perform much worse even though M_2 sends *more* FEC than our scheme.

Next, we examine how (65) behaves under changing $p(t)$ obtained from ns2 simulations. In this case, we vary the number of FTP connections N every 5 control intervals and measure long-term average packet loss at the receiver. The relationship between N and the long-term average loss is illustrated in Fig. 13(a) where the increase in packet loss is caused by the well-known TCP scalability properties [6]. Changing the value of N randomly over time, the network exhibits fluctuating packet loss shown in Fig. 13(b). Using this information, the sender estimates transition probabilities p_{00} and p_{11} for each interval and uses them in FEC control. Fig. 13(c) and (d) plot the evolution of U^H achieved by different FEC-control schemes and show that our adaptive controller exhibits behavior similar to that observed in Markov-loss simulations.

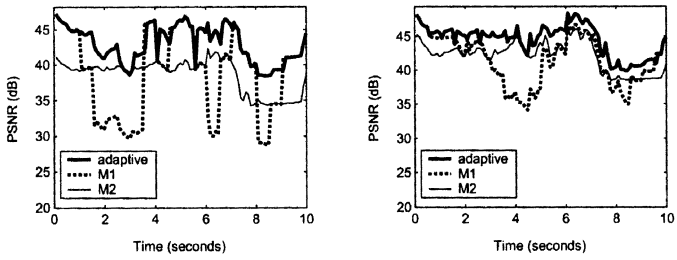
Analysis in previous sections suggests that both correlated and uncorrelated loss patterns, as well as exponential and heavy-tailed loss-burst lengths, lead to an almost identical behavior of U^H. Additional simulations with ns2 confirm that results based on simple Markov-loss models can indeed be used as first-step approximations to the real behavior of U^H in generic networks. Future work will examine this issue in more detail and attempt to understand how more complex loss patterns influence optimal selection of FEC overhead.

D. PSNR Quality

We finish the paper by comparing the adaptive method with fixed-FEC schemes using PSNR quality curves. We apply packet-loss information obtained through ns2 and Markov chain simulations to each MPEG-4 FGS frame of the Foreman video sequence. We enhance each base-layer frame using consecutively received FGS packets and plot PSNR quality curves accordingly. Note that for this comparison, we protected the entire base layer in all cases and allow random loss only in the FGS layer.

Fig. 14 plots PSNR curves for both simulation cases. Observe in Fig. 14(a) that M_1 suffers significant quality degradation when U^H drops around $t = 2$ seconds (see Fig. 12(a)). Similarly, M_2 exhibits suboptimal video quality during the entire streaming session due to its ψ being too large. Compared to the two cases M_1 and M_2, our adaptive method offers almost 6 dB higher PSNR than M_2 throughout the session and out-performs M_1 by almost 10 dB for half the duration of the streaming session.[6] Fig. 14(b) shows that the improvement in ns2 simulations is not as dramatic as that in the Markov example due to the

[6]Note that a 1-dB gain in PSNR is usually considered significant [19].



(a) Markov loss simulation (b) ns2 simulation

Fig. 14. PSNR of CIF Foreman reconstructed with different FEC overhead control.

lower packet loss rates, but nevertheless amounts to a 3–9 dB improvement.

VII. Conclusion

This paper studied the effect of random packet loss on scalable video traffic in best-effort networks and proposed an adaptive FEC overhead control mechanism that can provide high quality of video to end-users. We also investigated the characteristics of packet loss in an FEC block and derived practical models for the distribution of the number of lost packets in a block of fixed size under Markov packet loss. Furthermore, we examined several stochastic loss models for streaming video and conclusively established that proper control of FEC overhead can significantly improve the utility of received video over lossy channels.

References

[1] E. Altman, C. Barakat, and V. Ramos, "Queueing analysis of simple FEC schemes for IP telephony," in *Proc. IEEE INFOCOM*, 2001, pp. 796–804.

[2] S. Bajaj, L. Brelau, and S. Shenker, "Uniform versus priority dropping for layered video," in *Proc. ACM SIGCOMM*, 1998, pp. 131–143.

[3] E. Biersack, "Performance evaluation of forward error correction in ATM networks," in *Proc. ACM SIGCOMM*, 1992, pp. 248–257.

[4] P. Billingsley, *Probability and Measure*, 3rd ed. New York: Wiley, 1995.

[5] J. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive FEC-based error control for Internet telephony," in *Proc. IEEE INFOCOM*, Mar. 1999, pp. 1453–1460.

[6] A. Dhamdhere, H. Jiang, and C. Dovrolis, "Buffer sizing for congested Internet links," in *Proc. IEEE INFOCOM*, 2005, pp. 1072–1083.

[7] P. Frossard and O. Verscheure, "Joint source/FEC rate selection for qualtiy-optimal MPEG-2 video delivery," *IEEE Trans. Image Process.*, vol. 10, no. 12, pp. 1301–1304, Dec. 2001.

[8] E. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253–1265, Sep. 1960.

[9] D. Li and D. Cheriton, "Evaluating the utility of FEC with reliable multicast," in *Proc. IEEE ICNP*, 1999, pp. 97–105.

[10] D. Loguinov and H. Radha, "End-to-end Internet video traffic dynamics: Statistical study and analysis," in *Proc. IEEE INFOCOM*, 2002, pp. 723–732.

[11] Network Simulator (ns-2). [Online]. Available: http://www.isi.edu/nsnam/ns/

[12] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 2nd ed. New York: McGraw-Hill, 1984.

[13] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, no. 9, pp. 40–48, Sep. 1998.

[14] H. Radha, M. Schaar, and Y. Chen, "The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP," *IEEE Trans. Multimedia*, vol. 3, no. 3, pp. 53–68, Mar. 2001.

[15] P. Richards, *Manual of Mathematical Physics*. New York: Pergamon, 1959.

[16] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," in *Proc. 20th Annu. Conf. Local Computer Networks*, 1995.

[17] C. Stein, "A bound for the error in the normal approximation to the distribution of a sum of dependent random variables," in *Proc. 6th Berkeley Symp. Mathematical Statistics and Probability*, 1972, vol. 2, pp. 583–602.

[18] Suyono and J. Weide, "A method for computing total downtime distributions in repairable systems," *J. Appl. Probabil.*, vol. 40, no. 3, pp. 643–653, 2003.

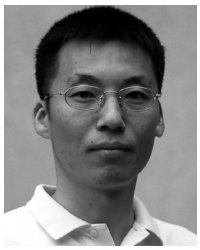
[19] M. Schaar and H. Radha, "Network and device driven motion-compensated scalable video for wireless systems," *Packet Video*, Apr. 2002.

[20] R. Wolff, *Stochastic Modeling and the Theory of Queues*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[21] H. Wu, M. Claypool, and R. Kinicki, "A model for MPEG with forward error correction and TCP-friendly bandwidth," in *Proc. NOSSDAV*, 2003, pp. 122–130.

[22] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *Proc. IEEE INFOCOM*, 1999, pp. 345–352.

[23] H. Yousefizadeh and H. Jafarkhani, "Statistical guarantee of QoS in communication networks with temporally correlated loss," in *Proc. IEEE GLOBECOM*, 2003, pp. 4039–4043.



Seong-Ryong Kang (S'04) received the B.S. degree in electrical engineering from Kyungpook National University, Daegu, Korea, in 1993, and the M.S. degree in electrical engineering from Texas A&M University, College Station, in 2000. He is currently working toward the Ph.D. degree in computer science at Texas A&M University.

During 1993–1997, he worked as a Patent Engineer/Consultant for First IPS, Korea. His research interests include media streaming, congestion control, and bandwidth estimation.

Dmitri Loguinov (S'99–M'03) received the B.S. degree (with honors) in computer science from Moscow State University, Moscow, Russia, in 1995, and the Ph.D. degree in computer science from the City University of New York, New York, in 2002.

Since 2002, he has been an Assistant Professor of computer science with Texas A&M University, College Station. His research interests include peer-to-peer networks, video streaming, congestion control, Internet measurement and modeling.