Modeling Application Based Forward Error Correction

James Westall
James Martin

School of Computing
Clemson University
Clemson, SC 29634-1906

September 24, 2010

1 Introduction

Forward error correction (FEC) can potentially provide improved performance in scenarios in which automatic repeat request (ARQ) protocols are impractical. Instances in which this is the case include transmission of real-time multimedia data where loss is undesirable, but retransmission delays are intolerable.

It is possible to implement FEC at the application layer. Such systems are sometimes called AL FEC or AP FEC. An AP FEC system that employs Reed-Solomon type erasure correcting codes necessarily uses fixed size packets. Short packets must be extended with some form of padding.

A FEC group consists of N packets with k of them being data packets and $N-k$ carrying redundant data. The coding efficiency of the FEC scheme is therefore k/N, and the redundancy is $(N-k)/N$. Data packets that are lost during transmission can be reconstructed if and only if a combination of at least k data and check packets of the N packet group are received correctly. When fewer than k packets of a group are received correctly, the data packets that are received can be added to the playback buffer, but any check packets that are received are simply discarded.

Historically, the Reed-Solomon coding system has been most widely used in FEC systems. Recently more efficient systems such as Raptor codes have gained considerable interest.

2 Modeling FEC performance

Measures of the effectiveness of a FEC system include the coding efficiency, k/N, and the packet loss rate. Packet loss rate is coupled to efficiency in the sense that reducing efficiency also reduces loss rate. Thus, it is desirable to find an operating point that provides both a reasonable efficiency and an acceptable loss rate. Further complicating the problem is the fact that in an operational network, the stochastic process representing the probability that an individual packet is loss is non-stationary and loss events are not independent.

2.1 Bernoulli loss process

The simplest form of packet loss process is the Bernoulli process. Bernoulli loss events are considered to be independent with each packet having probability of loss, p. In this case the probability of experiencing exactly j losses in a FEC group of N packets is binomial, $b(N,j,p) = \binom{N}{j} p^j (1-p)^{(N-j)}$.

Without FEC, the expected number of losses in an N packet FEC group is then

$$\sum_{j=1}^{N} j \binom{N}{j} p^j (1-p)^{(N-j)} = Np,$$

and thus the expected percentage of lost packets is $100p$.

When $N-k$ check packets are present and at least k are received correctly, there is no loss. That is, the probability of experiencing j losses for $j = \{1, 2, ..., N-k\}$ is 0. Therefore, the expected number of losses in this case

becomes

$$\sum_{j=N-k+1}^{N} j\binom{N}{j}p^j(1-p)^{(N-j)}$$

.

Unlike the previous case, there is no simple way to collapse this sum into a simple function of N, k, and p. Nevertheless, it is straightforward to compute it, and doing so provides some useful insights.

Let $r = (N-k)/N$ represent the redundancy of the code. It can be shown computationally that when the redundancy exceeds the loss probability, then the expected number of lost packets per FEC group monotonically approaches 0 as N approaches infinity. That is, if $r > p$, then

$$\lim_{N\to\infty} \sum_{j=N-k+1}^{N} j\binom{N}{j}p^j(1-p)^{(N-j)} = 0.$$

When the redundancy is less than the probability of loss, $r < p$, the expected number of lost packets per FEC group approaches Np as N approaches infinity in a non-monotonic way. That is, in contrast to the first case, increasing the size of the FEC group reduces FEC's effectiveness. In the limit the expected loss of Np is exactly what is obtained without the use of FEC.

In the third case, when $r = p$,

$$\lim_{N\to\infty} \sum_{j=N-k+1}^{N} j\binom{N}{j}p^j(1-p)^{(N-j)} = p/2.$$

In this final case the expected number of lost packets per FEC group monotonically approaches $p/2$ from above as N approaches infinity.

These three situations are illustrated in table 1. In this example $r = 0.2$ and the expected fraction of lost packets (including both data an check packets) is computed for $p \in \{0.199, 0.200, 0.201\}$ and for increasing values of N and k.

N	k	p	%loss
10	8	0.199	11.16
100	80	0.199	10.15
1000	800	0.199	9.45
10000	8000	0.199	8.02
100000	80000	0.199	4.27
1000000	800000	0.199	0.12
10	8	0.200	11.28
100	80	0.200	10.40
1000	800	0.200	10.13
10000	8000	0.200	10.04
100000	80000	0.200	10.01
1000000	800000	0.200	10.00
10	8	0.201	11.39
100	80	0.201	10.65
1000	800	0.201	10.81
10000	8000	0.201	12.07
100000	80000	0.201	15.79
1000000	800000	0.201	19.97

Table 1: Expected lost percentage

3 Characterizing the effectiveness of FEC

We begin by considering the effect of FEC on total loss rate. We will later address the important issue of distribution of losses.

Given a steady state packet loss rate of p in a k of N FEC system, if the FEC system is not successful in recovering any missing packets, then the expected loss fraction of for all packets is p and the expected loss fraction of data packets $p \times k/N$.

In the following examples, we will fix k/N at 0.8 and let N take values in the set $\{10, 20, 40, 80, 160, ..., 10240\}$. The redundancy of this scheme is 1 - 0.8 = 0.2. Therefore, we will restrict p to a range where increasing FEC group size is beneficial: $\{0.10, 0.12, ..., 0.18\}$. In the absence of FEC, the expected loss rate of data packets is $0.8p$ or $\{0.08, 0.10, 0.11, 0.13, 0.14\}$.

The loss rates achieved with an application FEC system in place are shown in figure 1. It can be noted that, since p is less than the redundancy, increasing N reduces the post-FEC loss rate. The reduction in loss rate appears approximately linear in the figure, but it should be noted that that a logarithmic scale is used on the x axis. Thus, the fraction of packets received correctly grows slowly, approximately as a function of $log(N)$.

We can look at this same data in another way by defining a measure of FEC effectiveness, $\epsilon = 1 - observed_loss_fraction/p$. If the observed loss fraction is p, then then $\epsilon = 0$. If the observed loss fraction is 0.0 then $\epsilon = 1$.. This data is shown in figure 2. Since this measure is normalized by the expected loss fraction p, it produces a more illuminating perspective on the effectiveness.

For example, when the FEC group size is 10 and the loss rate p is 0.18, the effectiveness is approximately 50% meaning that the observed total loss rate is approximately 9%, and thus the data packet loss rate is

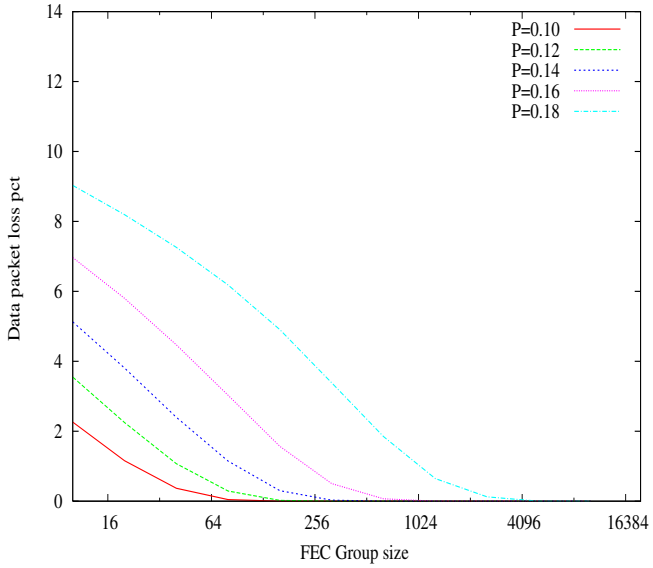


Figure 1: Loss rate for i.i.d. loss process

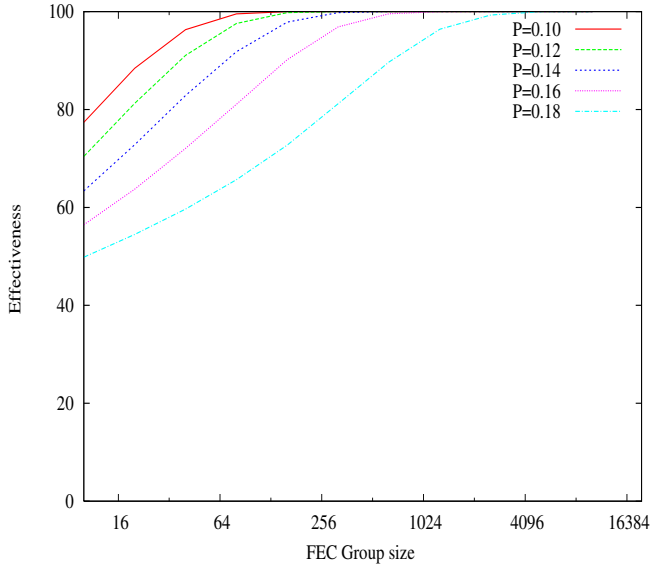


Figure 2: FEC effectiveness for i.i.d. loss process

$0.8 \times 9\%$ or 7.2

In contrast when the total loss rate is only 10% the FEC effectiveness is almost 80less than 2%.

4 Loss process models

It is the case that assuming the loss process is i.i.d. is not supported by real-world experiments and leads to optimistic predictions regarding the effectiveness of FEC. It has been established via measurement that actual loss processes tend to be bursty. Hence, we turn our attention to models that produce such bursty behavior.

4.1 Correlated loss models

In simulating a FEC system, the packet stream is modeled by a sequences of binary values where 0 represents a successful reception and 1 represents a packet loss. Thus an i.i.d. stream can be modeled by generating a stream of pseudo-random numbers that are uniformly distributed in $[0.0, 1.0)$ and mapping values in $[0.0, 0.5)$ to 0 and values in $(0.5, 1.0)$ to 1.

To model a correlated loss process, we seek a method of synthesizing a binary sequence, with prescribed mean and autocorrelation. The problem of generating pseudo-random numbers having a prescribed continuous distribution and autocorrelation has been studied by us and others, and efficient FFT based approaches are known.

The problem of generating a distribution of correlated binary values has also been widely studied. However, computationally efficient algorithms for generating long sequences with prescribed correlation are not known.

Nevertheless, it is the case that a simple and very efficient method for generating auto-regressive AR(1) sequences having autocorrelation parameter ρ was developed by Lunn and Davies [**?**]. They show that the algorithm below will produce a sequence with mean p and autocorrelation at lag n equal to ρ^n. Since the correlation decays exponentially, the resulting process is called *short-range dependent*.

```
u = unif();
if (unif() < rho)
   A = 1;
else
   A = 0;

u = unif();
if (unif() < mu)
   B = 1;
else
   B = 0;

Y = A * Yprev + (1 - A) * B;
Yprev = Y;
return(Y);
```

Results obtained for $\rho = 0.86$ are shown in figures 3 and 4. For small values of N, the efficiencies are now tightly clustered around 12% for all values of p. Since the correlation is short-range dependent, as N begins to

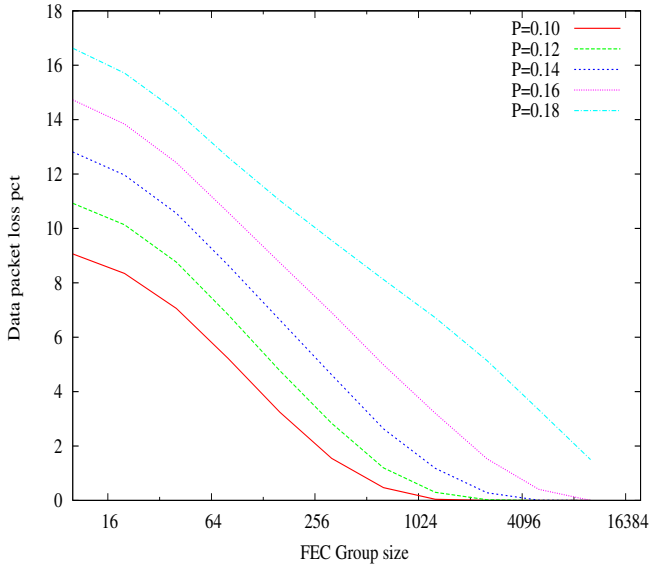


Figure 3: Loss rate: AR(1) loss process $\rho = 0.86$

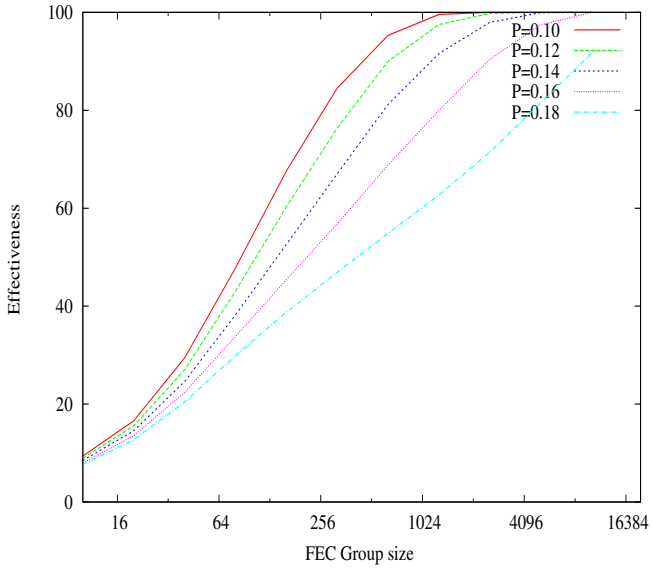


Figure 4: FEC effectiveness: AR(1) loss process $\rho = 0.86$

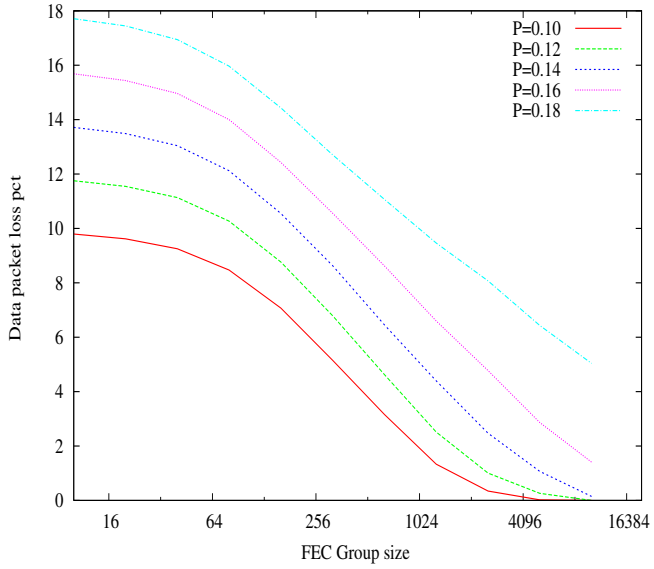


Figure 5: Loss rate: AR(1) loss process $\rho = 0.96$

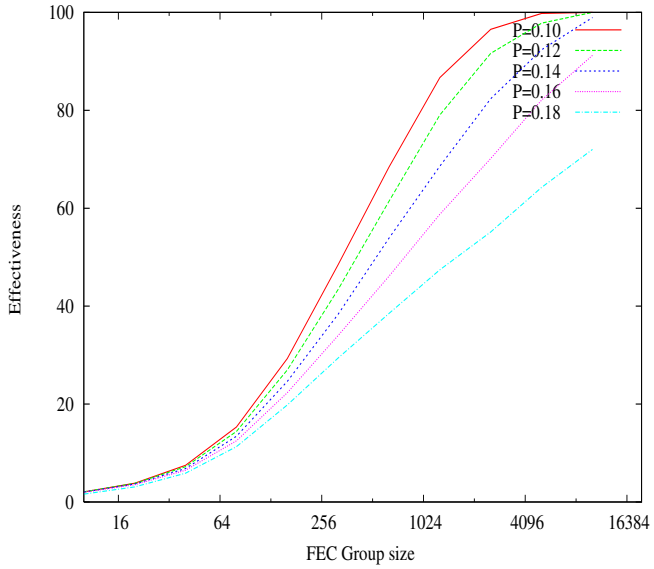


Figure 6: FEC effectiveness: AR(1) loss process $\rho = 0.96$

grow the efficiency curves disperse and it takes a FEC group size of 1280 to achieve a 50% FEC effectiveness at loss rate $p = 0.18$.

4.2 Hidden Markov models

In a hidden Markov model of the loss process, the system is assumed to be in exactly one of M states at any instant in time. Each state, i, has an associated loss probability p_i. Transitions between the states are defined by an MxM matrix $A = \{a_{i,j}\}$ where $a_{i,j}$ is the probability that if the system is currently operating in state i, it will transition to state j. Therefore, the values in each row of the matrix A must sum to 1.0.

The occurrence of state transitions occurs at fixed discrete times that may or may not be synchronous with packet transmission events. The matrix A should be defined so that starting in any state i it is possible (through a possibly large number of state transitions) to reach any other state j. In this case it is said that the discrete time Markov chain (DTMC) is irreducible.

In this case A^n (multiplying A by itself n times using standard matrix multiplication) converges to a matrix A' as n becomes large. When convergence occurs, all rows of A' will be identical and value of $a'_{i,j}$ will be the long-term fraction of time that is spent in state j.

Thus the global loss probability p will be:

$$p = \sum_{j=0}^{M-1} p_j a'_{0,j}$$

It should be obvious that the hidden Markov model allows one to define a virtually unlimited collection of loss behaviors.

4.2.1 Gilbert-Elliot Models

The Gilbert-Elliot models are widely used instances of the hidden Markov model genre whose use dates back to 1960 [**?**]. In these models $M = 2$ states, and the system consists of a 'good' state 0 with a low error rate and a 'bad' state 1 with a high error rate. State transitions are assumed to be synchronous with packet arrival events.

As described in the general case of hidden Markov models, the long-term packet loss probability is

$$p = p_0 a'_{0,0} + p_1 a'_{0,1}$$

Following Carvalho, we will represent $a_{0,0}$ by β and $a_{1,1}$ by α so that the state transition matrix is given by:

$$\begin{bmatrix} \beta & 1-\beta \\ 1-\alpha & \alpha \end{bmatrix}$$

For any given α and β and long-term loss rate p, it can be observed that the values $p_0 = 0.0$ and $p_1 = 1.0$ produce the most challenging environment for a FEC system. That is, while it is certainly possible to increase p_0 and make a corresponding decrease in p_1 that preserves p. The modified system will always have a lower post-FEC loss rate than the original. Since this is the case, we will assume that $p_0 = 0.0$ and $p_1 = 1.0$.

Given these choices of for p_0 and p_1, it remains to choose α and β. These values will determine both the long term loss probability p and the mean holding time in each state.

The holding time in a state of the DTMC is a Bernoulli process. That is, if the probability the process remains in the state is α, then the probability of leaving the state is $(1 - \alpha)$ and the mean holding time in the state is $1/(1 - \alpha)$.

Furthermore, if X represents the holding time in the state, then

$$Pr(X = n) = (1 - \alpha)\alpha^{(n-1)},$$

and it follows that the cumulative distribution function is given by:

$$Pr(X >= n) = \sum_{i=1}^{n}((1 - \alpha)\alpha^{(n-1)} = 1 - \alpha^n.$$

It can be noted that the density is a monotonically decreasing function meaning that a holding time of 1 is the most likely holding time regardless of the value of α. The density also decays exponentially with n. Nevertheless, as $\alpha \to 1$ and $(1 - \alpha) \to 0$ the density is smeared over quite a large range of values. For example, if $\alpha = 0.96$, then the mode of the holding time distribution is 1, the median is 17, the mean is 25, the 75th percentile is 34, and the 95th percentile is 74.

It should be clear that when a Gilbert-Elliot loss model is used, for a given long-term loss probability p, the effectiveness of FEC *decreases* as the mean holding time in the good and bad states increase.

4.2.2 Gilbert-Elliot Parameterization

We now show how to configure the Gilbert-Elliot parameters to produce an arbitrary long-term loss rate p and mean holding time in the 'bad' state.

Let $\gamma = 1 - \alpha$ and $\delta = 1 - \beta$. Then the mean holding time in the 'good' state is $1/\delta$ and in in the 'bad' state is $1/\gamma$. Because of the assumptions that $p_0 = 0$ and $p_1 = 1$, the long-term packet loss rate, p is thus

$$p = \frac{\frac{1}{\gamma}}{\frac{1}{\delta} + \frac{1}{\gamma}} = \frac{\delta}{\delta + \gamma} = \frac{1}{1 + \frac{\gamma}{\delta}}$$

From this, we see that to maintain a fixed value of p, the ratio of γ to δ must be a fixed value, r, that we can compute as:

$$r = \frac{(1-p)}{p}$$

.

Thus, for example, if $p = 0.2$ then $r = 4$ and so $\gamma = 0.8$ and $\delta = 0.2$ will provide the desired long-term loss rate of 0.2. However, with this configuration mean holding times in the good and bad state will be $1/\delta = 5$ and $\alpha = 1.25$ packet times respectively. These holding times are so short as to only marginally impact FEC and the results of simulating this loss model are virtually identical to those obtained with the i.i.d. model.

To increase holding times we must let γ and δ approach 0 while maintaining the 4 :: 1 ratio. For example, if we let $\gamma = 0.04$ and $\delta = 0.01$, then the mean holding time in the good state is 100 and in the bad state is 25.

The above analysis shows that the 0.0/1.0 Gilbert-Elliot model can be specified by two parameters: the long-term loss rate p; and the mean holding time in the bad state, $1/\gamma$. Given those values we can compute r as above and then compute $\delta = \gamma/r$ and then $\alpha = 1-\gamma$ and $\beta = 1-\delta$.

Loss and effectiveness measures for Gilbert-Elliot loss process having mean holding times of 8 and 25 packet times are show in figures 7, 8, 9, and 10.

Note how these results qualitatively approximate those of the AR(1) models with $\rho = 0.8$ and $\rho = 0.96$.

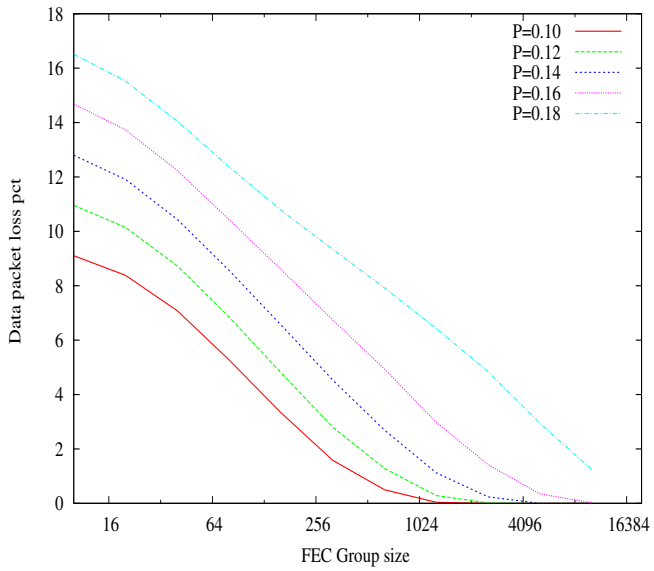


Figure 7: Gilbert-Elliot loss: mean holding time = 8

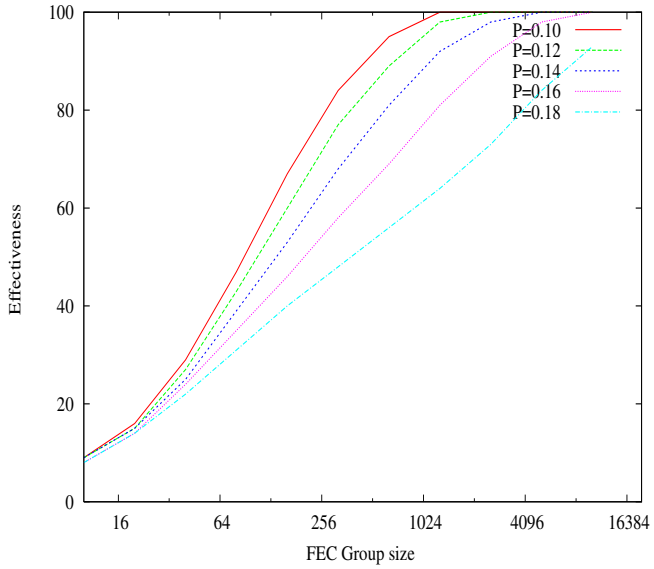


Figure 8: Gilbert-Elliot effectiveness: mean holding time = 8

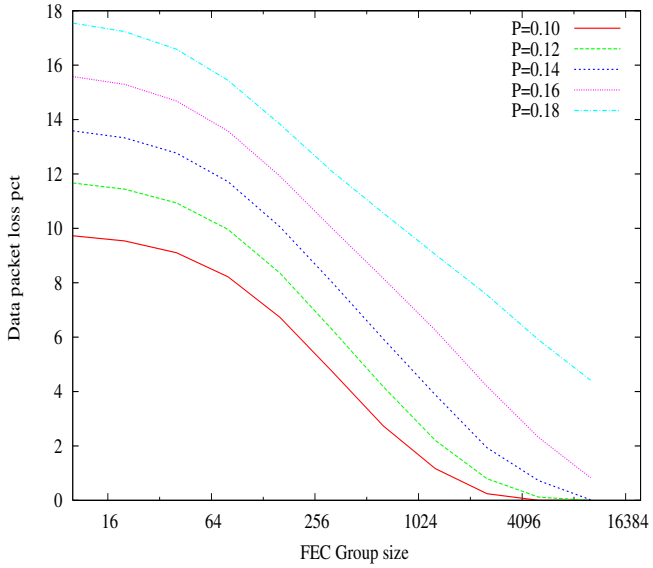


Figure 9: Gilbert-Elliot loss: mean holding time = 25

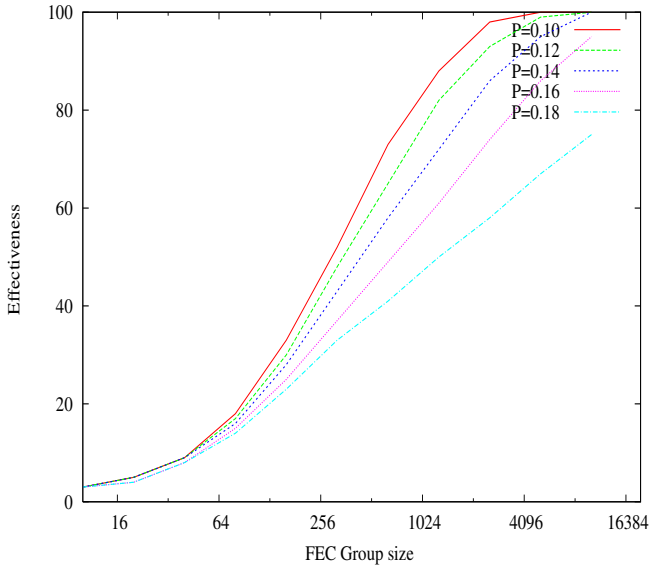


Figure 10: Gilbert-Elliot effectiveness: mean holding time = 25

4.3 Equivalence of AR(1) and G-E models

The qualitative similarity of the results obtained with similar parameterizations of the AR(1) and Gilbert-Elliot models led us to conjecture that the models are stochastically equivalent.

For the AR(1) model is is known that the autocorrelation,

$$R(n) = \rho^n$$

.

For the Gilbert-Elliot model we know that probability density function for the loss-burst length is:

$$Pr(X = n) = (1 - \alpha)\alpha^{(n-1)}$$

Thus we need to demonstrate that the autocorrelation of 0/1 sequence generated by the Gilbert-Elliot model is of the form $R(n) = \rho^n$, and that the loss-burst density of the AR(1) model is of the form

$$Pr(X = n) = (1 - \alpha)\alpha^{(n-1)}$$

It can be shown computationally and mathematically, that the loss burst density of the AR(1) process is

$$Pr(X = n) = (1 - p)(1 - \rho)(1 - (1 - p)(1 - \rho))^{n-1}.$$

Since $\gamma = 1 - \alpha$, The loss-burst density of the Gilbert-Elliot model

$$Pr(X = n) = (1 - \alpha)\alpha^{(n-1)},$$

may be reformulated as

$$Pr(X = n) = \gamma(1 - \gamma)^{(n-1)}.$$

From this it is apparent, that for equivalence of the two models to hold it is necessary that

$$\gamma = (1 - p)(1 - \rho).$$

Furthermore, we can solve the above equation for ρ and infer that for long-term loss rate p and mean loss-burst length $1/\gamma$, then the 0/1 process generated by the Gilbert-Elliot model will have an autocorrelation $R(k) = \rho^n$ where

$$\rho = 1 - \gamma/(1 - p)$$

.

From this, it is apparent that when lost burst size is fixed, ρ varies slowly as a function of p. For example, for $p = 0.1$ a mean loss burst of 25 corresponds to $\gamma = 0.04$ and $\rho = 0.96$. For, $p = 0.18$, $\rho = 0.95$.

4.4 Operating with $p \geq 1 - k/N$

It was previously pointed out that a FEC system has a discontinuity at the point at which $1.0 - k/N = p$. Figures 11 through 14 show the impact of operating with the loss probability greater than or equal to $1.0 - k/N$. Mean loss burst lengths of 8 and 25 are used as before.

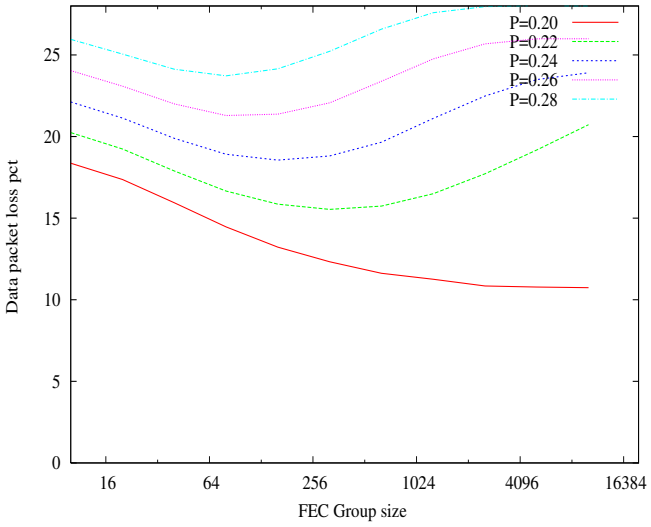


Figure 11: Gilbert-Elliot loss: mean holding time = 8

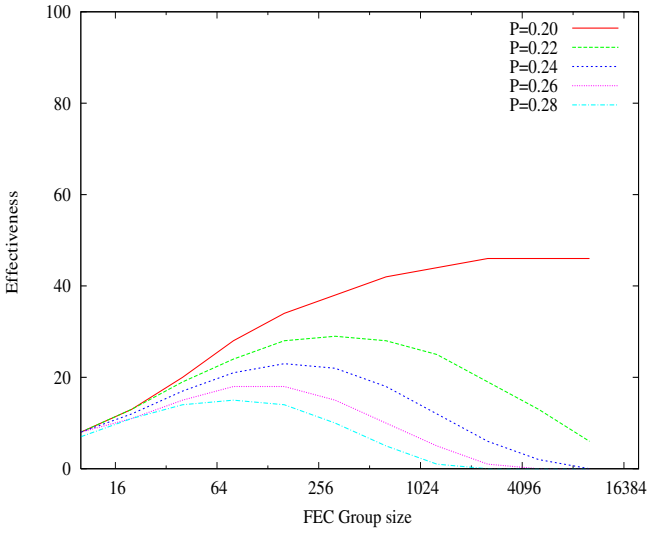


Figure 12: Gilbert-Elliot effectiveness: mean holding time = 8

The system is much less sensitive to both FEC group size and mean loss burst length in this domain. For $p > 1 - k/N$, data packet loss rate decreases for a time as group size increases then inevitably converges to the $p \times k/N$.

Even though performance is seen to be more stable in this operating domain, it should be noted that the stability represents a the inability of FEC to properly control loss!

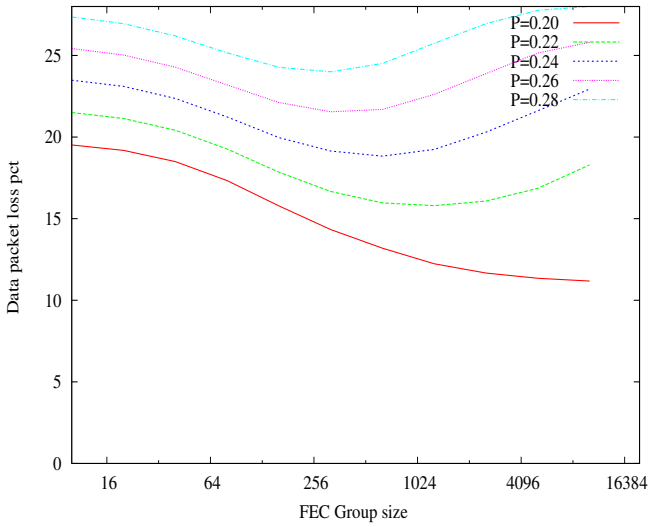


Figure 13: Gilbert-Elliot loss: mean holding time = 25

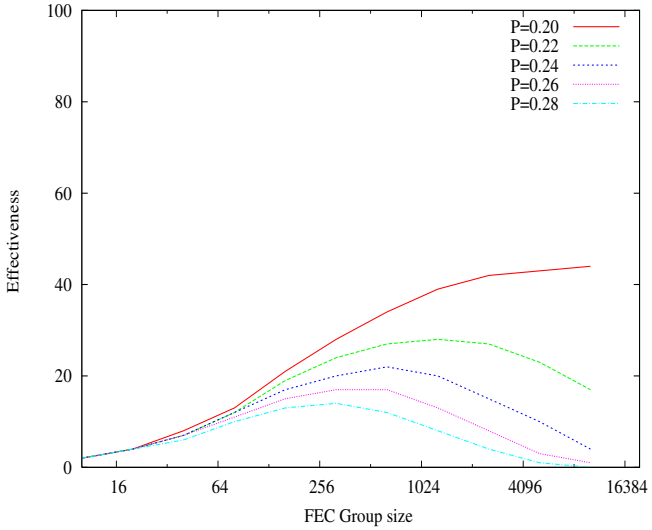


Figure 14: Gilbert-Elliot effectiveness: mean holding time = 25

4.5 Impact of high redundancy

It is shown in figure 15 when the redundancy is increased from the previous value of 0.2 to a new value of 0.6 the FEC effectiveness becomes nearly independent of the un-

derlying loss rate, p. As shown in figure 16, this doesn't mean that the observed loss rates are independent of p. It means that the ratio of observed loss rate to p is independent of p.

A beneficial aspect of this situation is that when group size reaches 320 loss rate is approximately zero for all values of $p \in [0.10, 0.18]$. It is doubtful that this serendipitous situation would persist as p approached the redundancy of 0.6, but it does reinforce the observation that as p values approach the redundancy, the FEC group size needed to sustain a near 0.0 loss rate becomes unbounded.

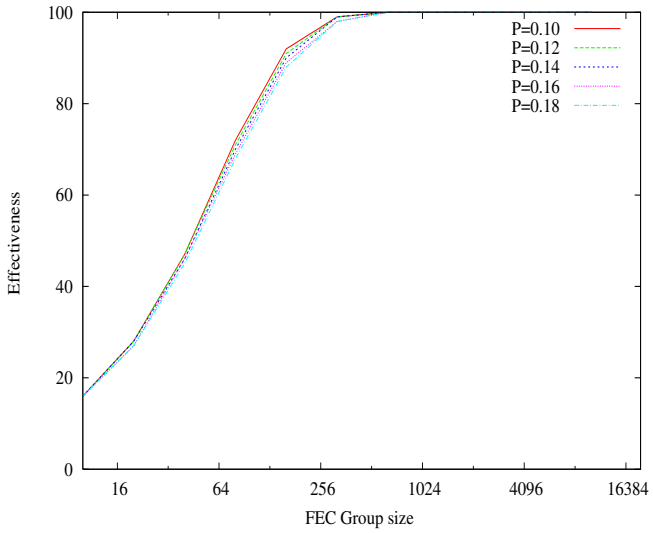


Figure 15: Gilbert-Elliot effectiveness: mean holding time = 25, 60% redundancy

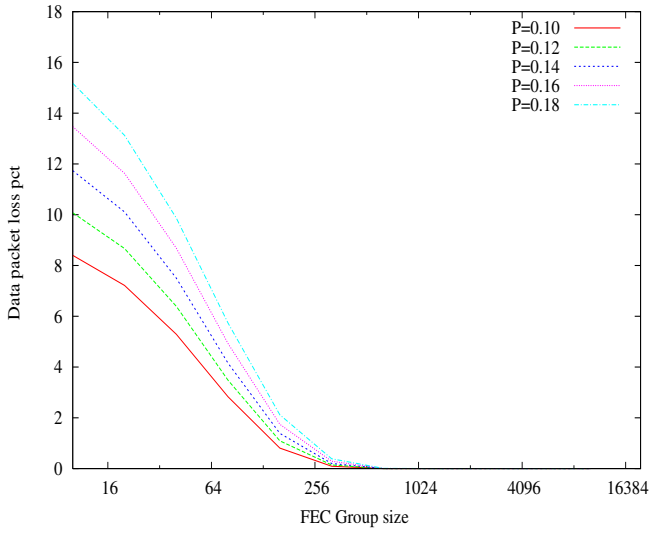


Figure 16: Gilbert-Elliot loss: mean holding time = 25, 60% redundancy

4.6 Packet length and FEC group size considerations

It is clear from the foregoing material that increasing FEC group size can significantly improve effectiveness if (and only if) the redundancy exceeds the long term loss rate. The disadvantages of large group sizes are: (1) potentially intolerable computational overhead in Reed-Solomon computations; and (2) extended latency of delivery to the playback buffer.

The first of these may be mitigated by the use of new and more efficient encodings such as Raptor. The second can be mitigated by using smaller packet sizes. For example, by using 300 byte packets instead of 1500 byte packets FEC group size can be increased by a factor of 5 with no latency cost and only a modest decrease in application throughput due to the increase in the number of headers that must be transported.

Thus, it might seem that simply reducing packet size would reduce loss rate. However, this perspective unfortunately depends upon an assumption regarding what is meant by mean loss burst length. In practice, the cause of a loss burst is likely to be a property of the physical environment that is properly measured in the *time domain* and not in the *packet count domain*.

Thus a mean loss burst length of 8 with 1500 byte packets does not correspond to a mean loss burst length of 8 with 300 byte packets. It corresponds more closely, but not exactly, to a mean loss burst length of 40 packets. A single physical event in the time domain that that kills 8 1500 byte packets kills all 40 300 byte packets only in the worst case. In the best case it would kill only 32.

Nevertheless, it is still not intuitively obvious how reducing packet size with the resulting larger FEC groups and corresponding longer mean loss burst lengths will impact loss rate. Hence, a simulation study was conducted. Table 2 shows the results of this study. Long term loss rate was fixed at $p = 0.1$ with redundancy at $r = 0.2$. The table shows the effect on data packet loss rate and data packet throughput of 4 successive halving of packet sizes and 4 corresponding doublings of FEC group size and mean loss burst length. The last column of the table shows the FEC effectiveness measure.

N	K	MLB	*loss*	*thpt*	*eff*
10	8	8	7.286	72.714	9
20	16	16	7.384	72.616	8
40	32	32	7.472	72.528	7
80	64	64	7.526	72.474	6
160	128	128	7.491	72.509	6

Table 2: Packet length effects

For these results, there is certainly no significant reduction in loss rate associated with reducing packet size and correspondingly increasing FEC group size and mean loss burst length. It fact it appears that there is a modest penalty.

However, these results don't tell the whole story. Suppose $p = 0.18$ and $\rho = 0.0$. Then $\gamma = 1.0 - 0.18 =$ and the mean loss burst length $1/\gamma = 1.222$. Since $\rho = 0.0$, this could be considered a Bernoulli loss process, and so there is no reason to claim that reducing the packet size has an impact on loss burst length. The results of these simulations are shown in table 3 they shown steady improvement.

N	K	MLB	$loss$	$thpt$	eff
10	8	1.222	7.245	72.755	50
20	16	1.222	6.574	73.426	54
40	32	1.222	5.853	74.147	59
80	64	1.222	5.001	74.999	65
160	128	1.222	3.906	76.094	73

Table 3: Uncorreleated losses

These two scenarios lead to an undesirable discontinuity in behavior as $\rho \to 0$. Hence we conclude that doubling the the mean loss burst length as we halve the packet length is unduly punitive. Instead we should seek to double only that portion of the lost burst length that is due to the positive correlation in the loss process.