Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality

Wenyu Jiang, Henning Schulzrinne
{wenyu,schulzrinne}@cs.columbia.edu
Department of Computer Science
Columbia University

Abstract

Internet packet loss and delay exhibits temporal dependency. If packet n is lost, packet $n+1$ is also likely to be lost. It leads to bursty network losses and late losses in real-time multimedia services such as Voice over IP (VoIP). This may degrade perceptual quality and the effectiveness of Forward Error Correction (FEC). To characterize this burstiness, we first discuss the modeling of packet loss and delay. We propose the joint use of the extended Gilbert model and the inter-loss distance (ILD) metric to characterize temporal loss dependency. For delay, we introduce a metric called the conditional cumulative distribution function. We have applied these models to some Internet packet traces to validate the necessity and effectiveness of these models. We then evaluate the effect of these dependencies on VoIP by investigating the final loss pattern (FLP) after applying playout delay adjustment and FEC. Our results through a set of simulations confirmed that the FLP is still bursty.

1 Introduction

Example of real-time multimedia applications include Voice over IP (VoIP), Internet radio stations, and video conferencing. The sender digitizes/encodes the media content and transmit it via the network as packets at regular intervals. The receiver gets the media packets and schedules an appropriate playout time in order to produce a smooth output media stream. It compensates for the delay variation (jitter) using a playout delay adjustment algorithm [7, 16, 14, 3, 2]. Simple algorithms use a fixed playout delay, either static or determined at the start of a session. More advanced VoIP applications compute a different playout delay for each talk-spurt [4] adaptively according to the current network condition.

The quality of multimedia applications is primarily determined by packet loss and delay. First, if a packet is lost, the media quality degrades unless there is a recovery mechanism such as Forward Error Correction (FEC) [15] [18] or retransmission. Second, if a packet delay is too high and misses the playout deadline, it leads to a late loss. Figure 1 illustrates how media encoding/decoding, FEC coding/recovery and playout delay adjustment work together in a typical VoIP application.

Undoubtedly, two-way metrics such as Round Trip Time (RTT) are important. In VoIP, a large RTT ($>$ 600 ms) will degrade the application's interactivity [5]. But as far as the receiver is concerned, the perceptual quality of what he/she received only underwent a one-way trip in the network. So we focus our analysis on one-way loss and delay. The distinction between one-way and two-way metrics blurs when the path characteristic between two ends are symmetric.

1.1 Why Burstiness Affects Quality

Packet loss and delay can exhibit temporal dependency or burstiness. For instance, if packet n has a large delay, packet $n+1$ is also likely to do so. This translates to burstiness in network losses and late losses, which may worsen the perceptual quality compared to random losses at the same average loss rate. In particular:

- It affects performance of FEC, e.g., percentage of packets that cannot be recovered. It is because FEC can recover a packet only if other necessary packets belonging to the same block are received.
- The loss pattern, whether the original one or the final losses after FEC, affects audio/video quality and effectiveness of loss concealment [9].
- To the end user, burstiness in late losses has no difference from network losses.
- Finally, as reported by Moon *et al.* [13] there is inter-dependency between delay and loss. It means late losses and network losses may merge into longer final loss bursts. This effect is shown later in this paper,

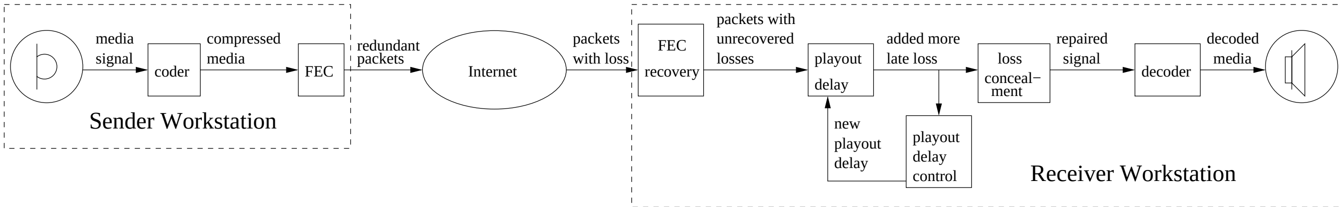

Figure 1: Loss recovery and concealment in packet audio

Simple metrics such as average loss and delay do not capture burstiness. Therefore our goal is to establish feasible metrics that can sufficiently characterize the packet loss and delay processes and reliably predict perceived quality. We propose a joint use of the extended Gilbert model [19] (2-state being a special case) and inter-loss distance (ILD) metric [12] to characterize temporal dependency in loss. The extended Gilbert model is suitable for describing loss run distributions, and the ILD metric is useful in capturing the burstiness between loss runs. To model temporal correlation of delay, we introduce a metric called conditional CDF. We analyzed some Internet packet traces and confirmed the existence of loss burstiness and delay correlation in these traces.

Past literature has focused on network loss patterns. In a real VoIP application, however, the perceptual quality is determined by the final loss patterns (FLPs) after playout delay control and optionally FEC. FEC can significantly change the loss rate and its distribution. The FLP could be even burstier due to the inter-dependency between delay and loss, that is, a packet loss is often preceded by high delays [13]. Finally, FEC coupled with playout delay adjustment complicates the FLP even further because a recovery time longer than the playout delay is equivalent to a late loss. We run a set of playout control simulations based on our Internet packet traces. The FLP is indeed bursty and also fits with the extended Gilbert model. The same is true when FEC is used.

There are four main contributions in our work: first, the joint use of the extended Gilbert model and the ILD metric. It is simpler than the nth-order Markov model (2^n states), yet depicts the burstiness both within and between loss runs. Second, we evaluate the errors introduced by the Bernoulli model, the 2-state Gilbert model and the extended Gilbert model when affected by small ILDs. The errors are computed over real Internet packet traces. Third, we introduce conditional CDF as a metric to capture temporal delay dependency. Finally, we have investigated the FLP after applying both playout control and optionally FEC. Our findings confirmed the FLP is still bursty.

Section 2 and 3 discuss the analytical modeling of loss and delay. Section 4 examines the final loss patterns for some packet traces after playout control and FEC.

2 Loss Modeling

It is generally agreed that of closely-spaced packets, packet losses are not approximated well by a Bernoulli model [19, 21, 2]. Since a packet loss in the Internet indicates congestion, the next packet may also be lost with a high probability, leading to the temporal loss dependency.

2.1 List of Network Packet Traces

Table 1 lists the Internet packet traces we used in this paper. To create the traces, two Unix workstations act as a sender and a receiver, respectively. The sender periodically transmit UDP packets of a fixed size. The software running on the workstations is not a VoIP application, but since its traffic is periodic, the trace can be used as if it were created by a real VoIP application. Many VoIP applications use silence suppression and introduces talk-spurts and gaps, we can mimic such behavior by randomly generating talk-spurts and gaps, and then omitting part of the trace where it maps to a gap. This is described in Section 4.1.

The UDP packet contains a sender timestamp and a sequence number. If the packet arrives, the receiver will record the arriving timestamp and write the two timestamps and sequence number into the trace file. Later during an off-line analysis, we calculate the one-way delays by subtracting the send/receive timestamps, followed by clock drift removal and initial clock difference correction. For simplicity, the initial clock difference is inferred assuming the network has symmetric delays. This assumption is not essential in our analysis since we are only interested in how delay variation (jitter) translates to late losses, and jitter is a relative measure instead of an absolute one. A packet loss is detected by a missing sequence number in the trace file.

In Table 1, each trace lists their average delay, jitter and loss rate. The average jitter is the arithmetic mean of the RTP jitter [20] (sec 6.3.1) for all arrived packets. It also gives a conditional loss rate to give a first glance of how bursty the losses are in a trace. The packet size listed in Table 1 is the UDP data portion in bytes. Spacing is the sender's inter-packet interval. Apparently, the smaller the spacing is, the stronger the temporal corre-

trace	sender	receiver	date	time	packets	$\overline{delay}$	$\overline{jitter}$	$\overline{loss}$	$\overline{clp}$	spacing	size
1	CU	GMD	9/19/1997	2:44pm	10039	57.6 ms	2.5 ms	0.47%	14.9%	30 ms	36 B
2	CU	UMass	9/19/1997	6pm	10978	62.9 ms	16.9 ms	9%	33%	30 ms	36 B
3	UCSC	CU	9/22/1997	1:30pm	10601	55.8 ms	5 ms	5.67%	10.6%	30 ms	36 B
4	UCSC	UMass	9/23/1997	8:12am	10290	56.8 ms	11 ms	2.82%	44.1%	30 ms	36 B
5	CU	UCSC	5/25/1999	5pm	12000	44.5 ms	1.9 ms	0.63%	14.7%	30 ms	36 B
6	CU	HP	6/1/2000	11:20am	50000	47.6 ms	1.8 ms	0.096%	31.3%	10 ms	36 B

Table 1: List of Internet packet traces being used

lation becomes. Most traces here use 30 ms, because it is the same as a frame duration in G.723.1 [10]. The frame size of G.723.1 at 6.3 kb/s is 24 bytes, which makes 36 bytes in a RTP/UDP packet. That is why most traces here also use 36 byte packet size.

Finally, all starting times in Table 1 are either Eastern Standard or Daylight savings Time (EST or EDT), whichever appropriate for the specified date.

2.2 The Gilbert Model

Sanneck *et al.* [19], Yajnik *et al.* [21] and Bolot *et al.* [2] recommend use of a Markov model to capture temporal loss dependency. All of them analyzed the 2-state Markov model, also known as the Gilbert model (Figure 2). It is simple to understand and to implement in monitoring applications.

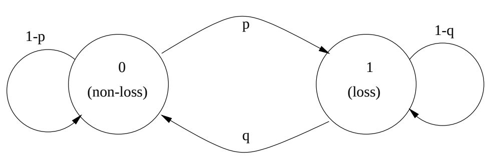

Figure 2: The Gilbert Model

In Figure 2, p is the probability that the next packet is lost, provided the previous one has arrived. q is the opposite. $1-q$ is the conditional loss probability (clp). Normally $p+q<1$. If $p+q=1$, the Gilbert model reduces to a Bernoulli model.

From the definition, we can compute π_0 and π_1, the state probability for state 0 and 1, respectively.

$$\pi_0 = \frac{q}{p+q}, \qquad \pi_1 = \frac{p}{p+q} \tag{1}$$

In the Gilbert model they also represent the mean arrival and loss probability, respectively.

p_k, the probability distribution of loss runs with respect to loss length k, has a geometric distribution: $p_k = (1-q)^{k-1} \cdot q$

To calculate p and q from a packet trace, one can use the loss length distribution statistics [19]. Let m_i, $i = 1, 2, ..., n-1$ denote the number of loss bursts having length i, where $n-1$ is the length of the longest loss bursts. Let m_0 denote the number of delivered packets.

$$p = (\sum_{i=1}^{n-1} m_i)/m_0, q = 1 - (\sum_{i=2}^{n-1} m_i \cdot (i-1))/(\sum_{i=1}^{n-1} m_i \cdot i)$$

As an example, take trace 1 in Table 1. It has the following loss burst distribution:

burst length	0	1	2	3
count	9992	34	5	1

$\pi_1 = (34 \cdot 1 + 5 \cdot 2 + 1 \cdot 3)/10039 = 0.0047$,
$p = (34+5+1)/9992 = 0.004$, $q = 0.851$

In this trace, the ulp is equal to $\pi_1 = 0.47\%$, whereas the clp is $1-q = 14.9\%$, significantly higher than π_1.

2.3 Approximation Error of the Bernoulli Model on a Gilbert Process

The Bernoulli model has only one parameter: the mean loss probability, $\hat{p}$. When used to approximate a Gilbert process, it would predict loss run distribution as: $\hat{p_k} = \hat{p}^{k-1} \cdot (1-\hat{p})$.

To illustrate, we compare trace 1 again. Here $\hat{p} = \pi_1 = 0.0047$. In Table 2, $\hat{n_k}, n_k$ are the expected number of loss runs with length k under the Bernoulli and Gilbert model, respectively.

loss length k	$\hat{p_k}$	$\hat{n_k}$	p_k	n_k
1	0.9953	39.8	0.851	34.0
2	0.00466	0.19	0.127	5.1
3	0.0000218	0.0009	0.019	0.76

Table 2: Estimation error caused by the Bernoulli model

For this trace it is evident that the Bernoulli model over-estimates single loss probability but under-estimates probability of longer loss bursts. Under the Bernoulli model, even double losses are highly unlikely for this trace, with an expected incident of 0.19, whereas the trace has 5 double losses.

2.4 General Markov Model

An nth-order Markov chain model is a more general model for capturing dependencies among events. The next event is assumed to be dependent on the last n events, so it needs 2^n states. Let X_i denote the binary event for ith packet, 1 for loss, 0 for non-loss. The parameters to be determined in an nth order Markov model are: $P[X_i|X_{i-1}, X_{i-2}, ..., X_{i-n}]$, for all combinations of $X_i, X_{i-1}, X_{i-2}, ..., X_{i-n}$.

Yajnik *et al.* [21] show that their packet traces typically have $n \leqslant 6$, and some require n to be 20 to 40. They did not quantify how much precision is gained by using an nth-order Markov model gains as compared to other simple models such as the 2-state Gilbert model.

2.5 Extended Gilbert Model

Sanneck *et al.* [19] proposes a different model that leads to fewer states. One only needs $n+1$ states to remember n events. It is called the *extended* Gilbert model. Their key distinction is that a general Markov model assumes all past n events can affect the future; whereas in an extended Gilbert model only the past (up to) n consecutive loss events will affect the future. Therefore, it does not capture the burstiness or clustering between loss runs. However, we can use the inter-loss distance metric [12] for this purpose.

Figure 3 illustrates how the extended Gilbert model works. The system keeps a counter l, which is the number of consecutively lost packets, but it is reset whenever the next packet is delivered. The parameter to determine in an extended Gilbert model is $P[X_i|X_{i-1}$ to X_{i-l} all lost], where X_i has the same definition as in the Markov model.

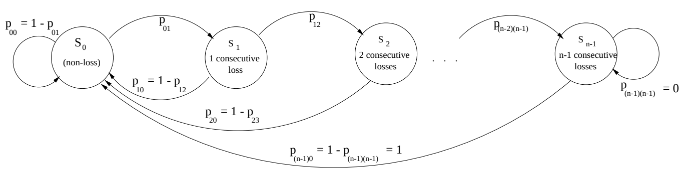

Figure 3: The extended Gilbert model

Therefore its corresponding transition matrix is:

$$P = \begin{bmatrix} p_{00} & p_{10} & p_{20} & \cdots & p_{(n-2)0} & p_{(n-1)0} \\ p_{01} & 0 & 0 & \cdots & 0 & 0 \\ 0 & p_{12} & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & p_{(n-2)(n-1)} & p_{(n-1)(n-1)} \end{bmatrix}$$

Therefore, its steady probability $(\pi_0, \pi_1, ...\pi_{(n-1)})$ can be calculated as follows:

$$P \times \begin{bmatrix} \pi_0 \\ \pi_1 \\ \cdots \\ \pi_{(n-1)} \end{bmatrix} = \begin{bmatrix} \pi_0 \\ \pi_1 \\ \cdots \\ \pi_{(n-1)} \end{bmatrix}, \quad \sum_{i=0}^{n-1} \pi_i = 1$$

It is now clear that the Gilbert model is a special case of the extended Gilbert model when $n = 2$.

[19] gives the formula to calculate the parameters for the extended Gilbert model, as follows:

$$p_{01} = (\sum_{i=1}^{n-1} m_i)/m_0 \qquad p_{(k-1)(k)} = (\sum_{i=k}^{n-1} m_i)/(\sum_{i=k-1}^{n-1} m_i) \tag{2}$$

where $k = 2, 3, ...n-1$, m_i is same as in section 2.2.

As an example, the parameters calculated from Trace 1 (CU-GMD) are: $p_{01} = 0.004$ $p_{12} = 0.15$ $p_{23} = 0.167$

Given a loss length distribution (length $\leqslant n-1$), an n-state extended Gilbert model completely retains the information of the distribution. It is because the transition matrix P has n unknowns and there are n equations in Formula 2 to determine the n unknowns.

As a comparison, below is the original loss length distribution for trace 2 (CU-UMass). The following table also lists what an equivalent 2-state Gilbert model produces on average. For this trace the Gilbert model predicts single and double losses quite closely, but the results become visibly different for $k > 2$. Generally we need to use the n-state extended Gilbert model to capture the original loss length distribution.

burst length k	0	1	2	3	4	5
trace count	9992	469	144	24	7	5
Gilbert model	9992	446	146	47.7	15.6	5.1

burst length k	6	7	8	9	10	11	12
trace count	8	2	0	1	1	1	1
Gilbert model	1.7	0.6	0.2	0.06	0	0	0

2.6 Inter-loss Distance Metric

The IPPM working group has proposed an inter-loss distance (ILD) metric [12] to describe the distance between packet losses in terms of sequence numbers.

The ILD metric is useful in two respects. First, the extended Gilbert model is able to model loss run distributions, but it does not model distances between loss runs. If many of the loss runs are close to each in sequence numbers, that is, they have small ILDs, it may also worsen the final perceptual quality. However, we need further subjective listening study to determine quantitatively how it relates to perceptual quality.

Second, small ILDs may also degrade the performance of FEC. Figure 4 shows the pmf of ILD for trace 4 from Table 1. In trace 4, about (5.5% + 5% + 6%)

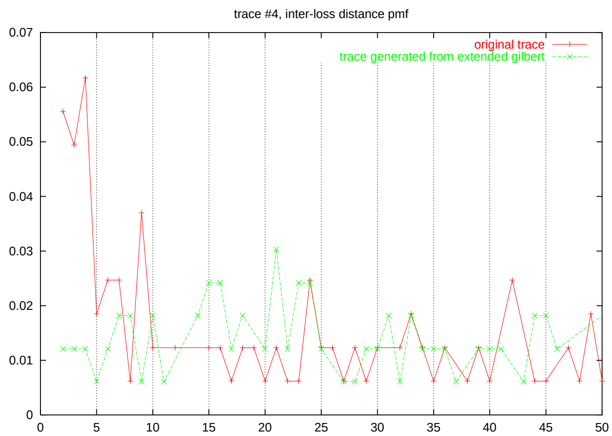

Figure 4: Inter-loss distance probability distribution

trace	original	gilbert	extended gilbert
4	44%	35%	40%

Table 3: Percentage of lost packets unrecoverable by FEC: effect of small ILDs on FEC

= 16.5% of the loss runs have an ILD < 5. However, if we generate a random loss trace using the extended Gilbert model, the probability of two loss runs having an ILD < 5 is less than 4% as inferred from Figure 4. Small ILDs has a direct effect on FEC performance, as illustrated in Table 3. Using a (5,3) Reed-Solomon FEC code [15], we compared the percentage of the lost packets that are unrecoverable by FEC. For example, 44% of the lost packets in trace 4 could not be recovered by FEC, but with an equivalent trace generated from the extended Gilbert model, the same ratio is 40%. Since the extended Gilbert model is the most detailed model for capturing loss run distributions, the only reason for this deviation is due to small ILDs in the original trace.

In section 4.4, we will examine the effect of ILDs on FEC for all the listed traces.

2.7 Other Loss Models and Metrics

IPPM working group has also defined a noticeable loss rate (NLR) metric [12]. Given a threshold distance d, one can compute the number of *noticeable* losses, that is, losses having an inter-loss distance $\leq d$. The noticeable loss rate is simply obtained by dividing the number of noticeable losses with the total number of packets.

NLR is useful in giving an estimate on how well FEC and loss concealment performs. But it does not capture burstiness within loss runs. For example, for NLR, a single loss run of 10 is equivalent to 5 double losses. The NLR is not exactly equivalent to the mechanism of FEC, since it does not have a notion of blocks. We have not yet examined the quantitative relationship between NLR and FEC performance.

3 Delay Models

3.1 Conditional CDF

One way to measure temporal delay dependency is by auto-correlation analysis. Let d_i denote the delay of ith packet, n the total number of packets measured, d the delay random variable, and $\bar{d}$ the average delay, l the correlation lag, the auto-correlation function (ACF) is:

$$\rho(d,l) = \frac{\sum_{i=1}^{n-l}(d_i - \bar{d})(d_{i+l} - \bar{d})}{\sum_{i=1}^{n}(d_i - \bar{d})^2} \quad (3)$$

The ACF is a good indicator of dependency, but it is difficult to calculate for example, the burst length distribution of late losses using this metric. Therefore, we introduce a new metric for this purpose: *conditional complementary* CDF, or just conditional CDF in short, defined as:

$$f(t) = P[d_i \geqslant t | d_{i-l} \geqslant t], l = 1, 2, 3, ..., \quad (4)$$

where l is the lag, t is the the x-axis in Figure 5. The formula means that if packet $i-1$ has a delay $\geqslant t$, then with probability $f(t)$ packet i will also have a delay $\geqslant t$.

We have found the conditional CDF (4) to be a simple and yet useful metric. This is because in real-time multimedia applications, any packet with a delay higher than the playout delay is effectively lost. By inspecting the unconditional CDF at a given playout delay D_p, the percentage of late (lost) packets is $1 - cdf(D_p)$. By inspecting the conditional CDF at D_p, we can estimate the burstiness of late losses. If the playout delay is constant throughout a session, the conditional CDF can be applied directly to estimate the burstiness of late losses. If an adaptive playout delay is used, we cannot directly relate the conditional CDF to late loss burstiness.

We have found little known literature on the topic of conditional delay dependency. Bolot [1] analyzed the conditional property of round-trip delays of consecutive packets. Their conclusion is that such delays have a random variation in lightly loaded conditions, and when background traffic load is high, consecutive delays often exhibit "spikes." A delay spike is a sequence of delays that starts with a high delay and then decreases almost linearly thereafter.

An example of conditional CDF is shown in Figure 5. It uses trace 1 in Table 1. The lag is 30 ms.

If the packet delays in our trace do not have significant temporal dependency, the conditional CDF at any lag l should look similar to the unconditional CDF. This is true in Figure 5 for low delays only. Beyond a certain

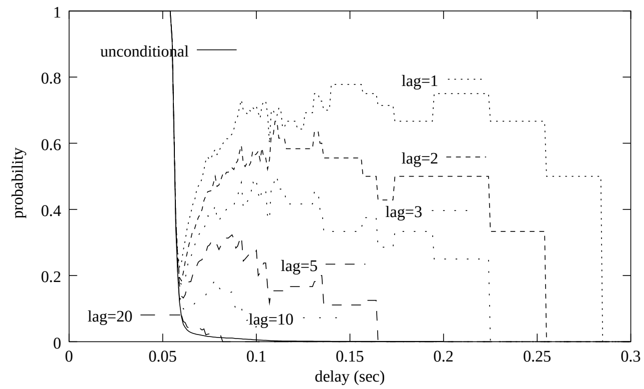

Figure 5: Conditional CDF for Trace 1 (CU - GMD)

threshold, the conditional CDF increases significantly. As lag l increases, the conditional CDF drops quickly, which is also confirmed by the ACF. But the ACF cannot tell us that only high delays have a strong temporal dependency.

The reason for the conditional CDF's increasing trend in Figure 5 can be explained intuitively as follows: a high delay for packet n indicates a non-empty router buffer. Since router operates at a limited speed, it takes some time for the buffer to drain. If the sender's inter-packet gap is small (e.g., 30 ms), the buffer depth may not have changed much, then the next packet will also likely experience a high delay. We also compute the queueing delay distribution in some approximated queueing models. Figure 6 shows the conditional CDF for an $M/D/1$ system with different lags [11].

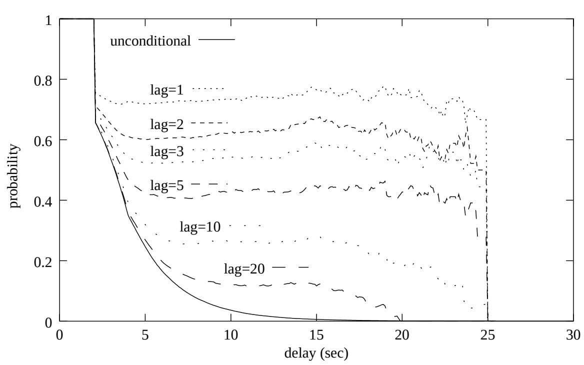

Figure 6: CCDF of $M/D/1$ system, $\lambda = \frac{1}{3}, \mu = \frac{1}{2}$

The temporal dependency of delay has a strong implication on the quality of real-time multimedia services. When a packet's delay exceeds a certain value (say the playout delay), the same will likely happen to the next packet. The result is burstier late-losses, which may degrade the performance of FEC [15] and could degrade the performance of loss concealment [9].

4 Effect of Delay and Loss on VoIP

Our ultimate goal is to predict end-user's perceived quality when using a real-time multimedia application, based on network performance. From Figure 1, we can see that the final quality perceived by an end-user depends on the final loss pattern (FLP) of the multimedia stream and the performance of loss concealment. Since the performance of loss concealment directly depends on the final loss pattern, we will investigate the FLP only. We assume loss recovery is done by FEC, but the analysis should be similar for a retransmission-based technique. The FEC we refer to is the traditional FEC, rather than a low bit-rate redundancy FEC as mentioned in [9]. A low bit-rate redundancy FEC would serve as a type of loss concealment in Figure 1.

4.1 Simulation Setup of Spurt/Gap Pattern, FEC, and Playout Control

VoIP applications often use silence suppressions to transmit only talk-spurts. Study of Brady [4] and Daigle [6] have found the spurt/gap distributions to be approximately exponential. We use an exponential distribution (1.5 sec average) plus a bottom threshold (240 ms) to describe the length distribution of both talk-spurts and silence gaps. The randomly generated spurts and gaps are then applied to an existing packet trace for playout control simulation. A packet is considered in the simulation only if its sequence number falls inside a talk-spurt. If that packet is lost in the original trace, then it is also considered lost in the simulation. It also means that on average half of the packet losses in the raw trace will not be considered because they don't fall inside a talk-spurt.

Our software simulates the (5,3) Reed-Solomon FEC code [15]. It uses a block size of 5 data units (i.e., packets). Among the 5 packets, 3 are original payload, 2 represent redundant information. As long as any 3 out of 5 packets are received, the payload can be fully reconstructed. It is the same FEC code that is used in [18]. When applying FEC to a VoIP packet stream, there is a choice of where to put the redundant information. Most applications piggy-back the redundant data onto a subsequent payload packet to reduce network load and packet header overhead. Furthermore, one needs to decide which "subsequent" packet to piggy-back on. We choose the settings as shown in Figure 7, because it is considered optimal in terms of correction ability [2].

We examine several playout control algorithms, the first is Exp-Avg, the exponential average algorithm in [16]. Its playout time p_i is calculated as follows:

$$p_i = t_i + \hat{d}_i + \mu \cdot \hat{v}_i$$

where i is the sequence number of the first packet in the current talk-spurt, t_i is the generation time of that

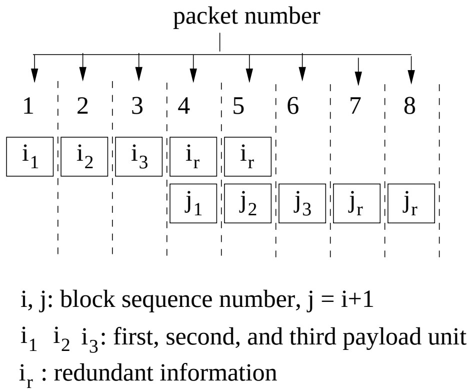

Figure 7: FEC settings

packet, and $\mu = 4$. $\hat{d_i}$ and $\hat{v_i}$ are the running averages of the delay and jitter, defined as:

$$\hat{v_i} = \alpha \cdot \hat{v_{i-1}} + (1-\alpha)|\hat{d_i} - n_i|$$
$$\hat{d_i} = \alpha \cdot \hat{d_{i-1}} + (1-\alpha)n_i.$$

where $\alpha = 0.998002$.

The second algorithm is the *virtualized* delay version of Exp-Avg with FEC, as mentioned in [18]. When FEC is used, if the playout algorithm is not aware of how long it needs to wait for FEC data to perform in-time recovery, the recovered data are often too late for playout. On the other hand, it is too conservative to always postpone playout for the whole FEC block to arrive when loss rate is very low. Rosenberg *et al.* [18] shows that any playout algorithm can be "virtualized" by taking the minimum of a normal packet's network delay and the delay when necessary FEC data arrive. The resulting minimum is called the virtual delay ($\hat{n_i}$), defined as:

$$\hat{n_i} = min(a_i, r_i) - s_i = min(a_i - s_i, r_i - s_i)$$
$$= min(n_i, r_i - s_i)$$

where a_i is the arrival time of packet i (or infinity if lost), s_i is its sending time, r_i is the earliest time that packet i can be fully recovered. For example, if packet 1 is the first unit in a (5,3) Reed-Solomon FEC block, and if it is lost, its earliest recovery time is when packet 4 arrives (assuming 2,3,4 are not lost and arrives in order).

To virtualize the Exp-Avg algorithm, all we need to do is replace n_i with $\hat{n_i}$ when updating $\hat{v_i}$ and $\hat{d_i}$. If the loss rate is high, the value of the virtual delay will be dominated by FEC recovery time. The "virtual" jitter will also increase because FEC recovery time is usually much higher than the average network delay. So the algorithm will increase the playout delay to allow most FEC recoveries to complete in time.

The third playout algorithm is Prev-Opt from [18], which is more adaptive than Exp-Avg. It calculates at the end of each talk-spurt, an optimal playout delay, D_{opt}. Then, it keeps track of a playout estimator D_w:

$$D_w = \rho \cdot D_{w-1} + (1-\rho) \cdot D_{opt}$$

where $\rho = 0.25$, and w is the spurt sequence number. The actual playout delay for the next spurt is:

$$D_{act} = D_w + \mu \cdot \hat{v_w}$$

where $\hat{v_w}$ is defined as:

$$\hat{v_w} = \alpha v_{w-1} + (1-\alpha)|D_w - D_{opt}|$$

In Prev-Opt, D_{opt} is defined as the optimal (i.e., miminum) playout delay that achieves (i.e., do not exceed) a user-specified application loss rate. For simplicity in analysis and implementation, we choose an application loss rate of 0%. In this special case, D_{opt} is simply the largest virtual delay of the last talk-spurt. This setting also reduces the number of late losses, because the D_{opt} is usually relatively large.

The fourth algorithm is the virtualized version of Prev-Opt when FEC is used.

4.2 FLPs after FEC and Playout Control

No FEC	(a) Exp-Avg					(c) Prev-Opt			
burst length	1	2	3	4	5	1	2	3	4
late-loss only	51	8	3	0	1	1	0	0	1
unrecovered loss	17	3	0	0	0	17	3	0	0
total loss	68	11	3	0	1	18	3	0	1
merged bursts	56	9	7	1	1	18	3	0	1

With FEC	(c) Exp-Avg					(d) Prev-Opt		
burst length	1	2	3	4	5	1	2	3
late-loss only	50	8	7	0	1	8	1	1
unrecovered loss	1	0	0	0	0	1	0	0
total loss	51	8	7	0	1	9	1	1
merged bursts	51	8	7	0	1	9	1	1

Table 4: Effect of playout control on final loss burstiness, CU-GMD trace, (5,3) Reed-Solomon FEC code

Table 4 is a brief summary of loss bursts for the CU-GMD Sep 1997 trace. The "unrecovered loss" is simply network packet losses if FEC is not used. If FEC is used, it is the number of loss-bursts that could not be recovered by FEC. The "merged bursts" column is the number of loss-bursts after merging late losses and unrecovered losses. For example, if a single late loss occurs at packet 37, and an unrecovered loss occurs at packet 38, then they form one loss burst of length 2, assuming no other packets are lost before or after them.

Due to the random nature of the generated spurt/gap patterns, the results listed in Table 4 is reproduceable only if the same implementation and random seed is used. But in most cases, the trends are similar independent of the random seeds. That is, the FLPs are bursty and we need the extended Gilbert model to describe such patterns.

In Table 4 (a) there are 17 single network losses and 51 single late losses. In Table 4 (b), only 1 out of 17 single network losses could not be recovered by FEC. That means the FEC does a good job of loss recovery, but since the loss rate is already low, and the jitter of this trace is also low, most of the recovered packets are not played back in time and become late losses.

burst length	1	2	3	4	5	6	7	11
late-loss only	7	4	0	0	0	0	0	0
unrecovered loss	223	62	10	4	2	3	2	1
total loss	230	66	10	4	2	3	2	1
merged bursts	227	66	10	4	1	2	4	1

(a) Without FEC, using Exp-Avg playout

burst length	1	2	3	4	5	6	7	10
late-loss only	25	8	0	0	0	0	0	0
unrecovered loss	22	10	5	5	2	4	0	1
total loss	47	18	5	5	2	4	0	1
merged bursts	48	18	6	5	2	3	1	1

(b) With FEC, using (5,3) Reed-Solomon code and virtualized Exp-Avg

burst length	1	2	3	4	5	6	7	11
late-loss only	0	0	0	0	0	0	0	0
unrecovered loss	223	62	10	4	2	3	2	1
total loss	223	62	10	4	2	3	2	1
merged bursts	223	62	10	4	2	3	2	1

(c) Without FEC, using Prev-Opt

burst length	1	2	3	4	5	6	10
late-loss only	0	0	0	0	0	0	0
unrecovered loss	22	10	5	5	2	4	1
total loss	22	10	5	5	2	4	1
merged bursts	22	10	5	5	2	4	1

(d) With FEC, using (5,3) Reed-Solomon code and virtualized Prev-Opt

Table 5: Effect of playout control on final loss burstiness, CU-UMass Sep 1997 trace

Also, if the late losses and network losses are adjacent in sequence numbers, they merge into bigger loss bursts. This is evident in the last column of Table 4 (a), where number of triple losses increased from 3 to 7. This effect is much less visible in Table 4 (b), because there are not many unrecovered losses to begin with.

We have performed the same experiment on other network traces we obtained. Table 5 is the result for trace #2 (CU-UMass). The results are in general similar: the FLPs are still best described by an n-state extended Gilbert model, and usually $n > 2$. But the effects of merging between unrecoverd and late losses are less evident. This is because the other traces have a much larger jitter, and hence a more conservative (larger) playout delay. The end result is there are far fewer late losses in these traces.

The conclusion we draw here is: after applying playout control and possibly FEC, the FLP is still best described an n-state extended Gilbert model. Our results indicate that usually $n > 2$. FEC generally does a good job of recovering network losses, but whether recovery is timely depends on the playout delay algorithm. There is also a merging effect between late losses and unrecovered losses. This effect, however, is minimized when both delay jitter is high and FEC is employed, which leads to a more conservative (higher) playout delay and recovery of most lost packets.

4.3 Comparisons of Delays Introduced by FEC

Table 6 compares the average playout delay between different playout algorithms. It also lists the clp in the FLP to give a first glance of its burstiness.

The delays are in ms. The first value before the '/' is the actual playout delay, the value after the '/' is the optimal (minimum) delay achievable at the same loss rate. The unconditional loss probability (ulp) and its conditional loss probability (clp) are listed as a '/' pair in the table. We can see that all traces exhibit a high clp in the FLP, which indicates a high degree of burstiness.

In Table 6, The FEC (i.e., virtualized) version of Exp-Avg does not add significant playout delay compared to the plain Exp-Avg. However, the gain of FEC is also relatively small because the unconditional loss probability is reduced significantly. Prev-Opt with FEC performs much better in terms of losses, but it also adds a large overhead the average playout delay. And we can see that Prev-Opt with FEC produces relatively conservative (i.e., large) playout delay compared to its optimal value. This is likely due to the choice of 0% application loss rate in our Prev-Opt implementation.

4.4 Error of Gilbert Model in Predicting FEC Performance

We have also investigated the error introduced by the Gilbert model when it is used to predict FEC performance. We first compute the two parameters needed in Gilbert model: p and q, which can be derived from ulp and clp. Then a program generates a packet trace with a Gilbert loss pattern. Next, we run the same FEC/playout simulation program on the generated trace. The program records two numbers: the number of original packet losses and the number of packet losses unrecoverable by FEC. Then we compare the percentage of unrecoverable packets for both traces. To minimize the error and variance due to random sampling, we run the simulations many times to obtain an average. The results have a large standard deviation (not shown here),

trace	FEC,Exp-Avg		no FEC,Exp-Avg		FEC,Prev-Opt		no FEC,Prev-Opt	
	delay/opt	ulp/clp	delay/opt	ulp/clp	delay/opt	ulp/clp	delay/opt	ulp/clp
1	82.6/68.4	1.6%/34%	80.2/66.5	2%/25%	156.6/85.0	0.32%/42%	140.4/79.5	0.56%/30%
2	248.1/168.4	3.9%/47%	236.9/134.1	9.3%/33%	542.9/186.5	2.7%/58%	415.2/146.4	9%/32%
3	108.4/90.4	4.2%/21%	92.1/70.5	6.2%/14%	387.1/149.7	0.48%/43%	148.0/77.4	6.1%/18%
4	134.9/108.4	2.8%/52%	128.0/101.0	3.8%/41%	402.1/157.2	1.6%/73%	318.2/121.6	2.9%/48%
5	62.3/51.8	1.3%/16%	60.4/49.8	2.0%/16%	105.6/60.6	0.37%/45%	88.1/53.8	0.73%/23%
6	59.1/53.8	2.3%/46%	58.9/53.7	2.4%/45%	81.8/55.2	0.28%/54%	81.7/55.2	0.28%/58%

Table 6: Average playout delays and conditional loss probability between different algorithms

trace	original	gilbert	extended gilbert
1	9.1%	8.4%	8.6%
2	28%	26%	25%
3	9.2%	7.8%	8.7%
4	44%	35%	40%
5	15%	7.5%	14%
6	43%	18%	16%

Table 7: Percentage of lost packets unrecoverable by FEC: effect of small ILDs on FEC, all traces

which we believe it is due to the small number of packet losses and relatively short length of our traces. The average still shows a consistent performance difference between different loss models.

Table 7 lists the FEC performance results for all the traces in Table 1. It also lists the results for the extended Gilbert model. In that case, a similar random trace is generated, except using the extended Gilbert loss pattern. Since the extended Gilbert model is the most detailed model for describing loss run distributions, its performance difference with the original trace is an indication of small ILDs (inter-loss distances) as explained in Section 2.6.

According to Table 7, trace 1 has similar results for all three columns, with the percentage for the original trace being a bit higher. The same is true for trace 2 and 3. Trace 4 shows a higher deviation between the simple and extended Gilbert model (35% vs. 40%), meaning that the Gilbert model is less accurate for this trace. There is also a large difference between the extended Gilbert model and the original trace (44%), which has to do with the small ILDs in trace 4. Recall from section 2.6 that in trace 4 about 16.5% loss runs has an ILD < 5. In trace 6, 36% of the loss runs have an ILD $\leqslant 3$ (ILD distribution not shown here), therefore the effect of small ILDs is even stronger. Finally, trace 5 has a large difference between the two Gilbert models, but there is almost no difference between the extended Gilbert model and the original trace.

4.5 Further Study: Effect of FLP on VoIP Subjective Quality

To summarize, the final loss pattern after playout adjustment is burstier than one would have expected. How this affects end-user perceptual quality requires further study. Rosenberg [17] has reported that the built-in loss concealment mechanism of G.729 codec can usually repair a single loss well, but does not work well on longer bursts. Therefore, with the same loss probability, a burstier loss pattern could degrade a voice signal to a greater degree than random losses, but there might well be exceptions. For example, when audio packet duration is very short (e.g., 5 ms) and average loss rate is high, random losses translate into a frequent annoying clicking sound (assuming no loss concealment). If the losses were bursty, it may translate into a less frequent clicking sound and become less annoying. A different example is for video streams. A video frame often consists of several network packets, and losing one packet renders the whole frame useless. In such case bursty losses may actually be preferable to random losses [8].

5 Conclusion

We discussed factors affecting real-time multimedia QoS. The first is the modeling of network delay and loss. We propose the joint use of the n-state extended Gilbert model and inter-loss loss distance (ILD) to characterize loss burstiness. This is confirmed by comparing the errors in estimating FEC performance between the simple and extended Gilbert model, and the original packet trace. We introduce the conditional CDF to capture the temporal dependency in network delays, that is, when previous delays are high, the next delay is also likely to be high. applying playout delay adjustment and optionally FEC, we have found that the final loss pattern (FLP) is still burstier than random losses and needs to be described by the extended Gilbert model. Particularly if FEC is not employed and jitter is low, late losses and network losses often merge into longer loss bursts. It is due to the observed inter-dependency between loss and delay, e.g., a loss is often preceded by high delays.

6 Future Work

We plan to perform subjective listener tests to examine how loss burstiness relates to perceptual quality. So far we have assumed the FLP is what determines perceptual quality. However, an algorithm such as Prev-Opt can produce swiftly changing playout delays, the resulting talk-spurts could be artificially squeezed or pulled, which may make the audio less comprehensible to the end user. So we also plan to investigate the effect of final playout jitter on perceptual quality.

7 Acknowledgements

We would like to thank Jonathan Rosenberg for providing some of the network packet traces and the measurement software tool. We also thank Prof. J.J. Garcia and Brad Smith of UC Santa Cruz and Dr. Puneet Sharma at HP Labs for providing us measurement accounts.

References

[1] Jean-Chrysostome Bolot. Characterizing end-to-end packet delay and loss in the Internet. *Journal of High Speed Networks*, 2(3):305–323, 1993.

[2] Jean-Chrysostome Bolot, Sacha Fosse-Parisis, and Don Towsley. Adaptive FEC-Based error control for interactive audio in the internet. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, New York, March 1999.

[3] Jean-Chrysostome Bolot and Andres Vega Garcia. Control mechanisms for packet audio in the internet. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, San Fransisco, California, March 1996.

[4] Paul T. Brady. A technique for investigating on-off patterns of speech. *Bell System Technical Journal*, 44(1):1 – 22, January 1965.

[5] Paul T. Brady. Effects of transmission delay on conversational behavior on echo-free telephone circuits. *Bell System Technical Journal*, 50(1):115–134, January 1971.

[6] John N. Daigle and Joseph D. Langford. Models for analysis of packet voice communications systems. *IEEE Journal on Selected Areas in Communications*, SAC-4(6):847–855, September 1986.

[7] Prabandham M. Gopal, J. W. Wong, and J. C. Majithia. Analysis of playout strategies for voice transmission using packet switching techniques. *Performance Evaluation*, 4(1):11–18, February 1984.

[8] Roch Gurin and Henning Schulzrinne. Network quality of service. In Ian Foster and C. Kesselman, editors, *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, San Francisco, California, 1998.

[9] Vicky Hardman, Angela Sasse, Mark Handley, and Anna Watson. Reliable audio for use over the internet. In *Proc. of INET'95*, Honolulu, Hawaii, June 1995.

[10] International Telecommunication Union. Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s. Recommendation G.723.1, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, March 1996.

[11] Wenyu Jiang and Henning Schulzrinne. Qos measurement of internet real-time multimedia services. Technical Report CUCS-015-99, Columbia University, New York, New York, December 1999.

[12] R. Koodli and R. Ravikanth. One-way loss pattern sample metrics. Internet Draft, Internet Engineering Task Force, June 1999. Work in progress.

[13] Sue B. Moon, Jim Kurose, Paul Skelly, and Don Towsley. Correlation of packet delay and loss in the internet. Technical report, University of Massachusetts, January 1998.

[14] Sue B. Moon, Jim Kurose, and Don Towsley. Packet audio playout delay adjustment algorithms: performance bounds and algorithms. Research report, Department of Computer Science, University of Massachusetts at Amherst, Amherst, Massachusetts, August 1995.

[15] Colin Perkins, Orion Hodson, and Vicky Hardman. A survey of packet loss recovery techniques for streaming audio. *IEEE Network*, 12(5):40–48, September 1998.

[16] Ramachandran Ramjee, Jim Kurose, Don Towsley, and Henning Schulzrinne. Adaptive playout mechanisms for packetized audio applications in wide-area networks. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 680–688, Toronto, Canada, June 1994. IEEE Computer Society Press, Los Alamitos, California.

[17] Jonathan D. Rosenberg. G.729 error recovery for internet telephony. Technical report, Columbia University, 1997.

[18] Jonathan D. Rosenberg, Lili Qiu, and Henning Schulzrinne. Integrating packet fec into adaptive voice playout buffer algorithms on the internet. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, 2000.

[19] H. Sanneck, G. Carle, and R. Koodli. A framework model for packet loss metrics based on loss runlengths. In *SPIE/ACM SIGMM Multimedia Computing and Networking Conference*, January 2000.

[20] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: a transport protocol for real-time applications. Request for Comments 1889, Internet Engineering Task Force, January 1996.

[21] Maya Yajnik, Sue Moon, Jim Kurose, and Don Towsley. Measurement and modelling of the temporal dependence in packet loss. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, New York, March 1999.