A method for predicting packet losses with applications to continuous media streaming*

Fernando J. Silveira Filho and Edmundo de Souza e Silva

Programa de Engenharia de Sistemas e Computação – COPPE/UFRJ
Caixa Postal: 68.511 – 21941-972 – Rio de Janeiro, RJ – Brasil

{fernando,edmundo}@land.ufrj.br

Abstract. *There is a number of applications that can benefit from the estimation of packet loss statistics. For instance, suppose that the loss process characteristics in an end-to-end path can be well approximated in advance. Then, a real time audio or video streaming application could adapt its transmission rate and choose the appropriate packet loss recovery strategy in order to deliver data with an acceptable quality. Adaptive mechanisms should be sufficiently accurate to capture the relevant loss measures and yet simple enough to be used in a real time protocol. In this paper, we evaluate different hidden Markov chain based models as predictors of short-term loss statistics. We propose an adaptive algorithm to estimate near future losses based on recent measurements and compare the accuracy of different underlying models.*

1 Introduction

There has been a growing interest in adaptive network protocols for tasks such as multimedia traffic rate control, path-switching and packet loss recovery mechanisms to name a few. Such mechanisms must be capable of inferring future packet losses and self-adjust their behavior in order to cope with the variability in the network conditions aiming at achieving some given performance goals ([Bolot et al. 1999], [Duarte et al. 2003], [Karol et al. 2004], [Tao and Guerin 2004], [Tao et al. 2005]). These control mechanisms often rely on packet loss models that need to be accurate and yet simple enough for real-time analysis. Unfortunately, the exact dynamics of packet loss processes in the Internet can be exceptionally different both across space - i.e. across different end-to-end paths - and in time, within the same path ([Zhang et al. 2000]), which adds to the difficulty of inferring losses.

Differences in traffic demands and capacities across links amount for the Internet's complex and unpredictable nature. The more recent emergence of wireless technologies add to this scenario the inherent unreliability of its transmission medium, where bit error rates are ordinarily many times higher than those seen on wires. In addition, it is not uncommon to find measurements that exhibit some sort of non-stationary phenomena like long-term periodicity or trends in the average loss rate which are difficult to model and even worse to track in real-time. However, what is perhaps most noteworthy is the fact that, even on channels whose statistics remain stationary over time, one can find indications of significant correlation between packets that are separated by up to a second ([Yajnik et al. 1999], [Duarte et al. 2003]).

*This work was supported by grants from CNPq and FAPERJ and a fellowship from CAPES.

For those reasons, it is of major importance not only to build *simple* and *flexible* packet loss models, but also to design strategies that allow an adaptive protocol to accurately predict future packet loss statistics while accounting for the effects of recent losses. By *simple*, we mean that a model should be computationally tractable to be usable by the application. *Flexibility*, on the other hand, implies that a model should do its best to fit a reasonable set of observed features from the real process. More importantly for our purposes, a model should be able to adapt to changes in the loss process overtime and predict future performance, conditioned on these local effects.

Probably the most applied models for packet loss processes are the Bernoulli process and the 2-state Markov chain, usually referred to as Gilbert model. Recently, the more general hidden Markov models have also emerged in the context of loss modeling ([Salamatian and Vaton 2001]). Evaluating the accuracy of these models to predict losses is one of the central issues discussed in this paper.

We attempt to explore further the use of hidden Markov models as a tool to predict losses. We develop a novel recursive algorithm that evaluates the distribution of the number of packets lost in a time window in the future given the recent history. We assess the goodness of such predictions using different hidden Markov models. We also propose a variation of the basic HMM approach, by constraining the model structure. Our model structure has two nice properties. First, by restricting the model we aim at reducing the total number of parameters to be estimated, thus lowering the overall complexity of the model estimation phase. Second, by assuming such specific patterns in the set of parameters, we attempt to capture the short-term dependencies in packet loss events with a Gilbert model while the longer-term dynamics is governed by a hidden Markov chain.

We survey the related literature and review some basic concepts in section 2. In section 3 we present the measurement experiment performed in order to collect packet loss data. Section 4 introduces the proposed prediction algorithm and the methodology used in our experiments. In section 5, we present qualitative results of the method applied to models. We then propose a new model, in section 6, which we consider better suited to handle the task of packet loss prediction and present, in section 7, its performance using the aforementioned algorithm. Section 8 develops on the computational costs of the prediction task. Finally, section 9 concludes the paper by summarizing our results and discussing the future directions for this research.

2 Related Work and Background

A simple and largely applied tool for modeling packet loss is the 2-state Markov chain, usually referred in the literature as a Gilbert or Gilbert-Elliot model ([Elliott 1965]). More recently, [Su et al. 2004] developed an iteration for the Gilbert model that allows the evaluation of the probability of observing i errors out of j transmissions, $P(i, j)$, conditioned on loss rate feedback from the channel. In our paper, we present an algorithm that computes the $P(i, j)$ conditioned on recent measurements for general hidden Markov models. Although our procedure can be applied to Gilbert models as well, it is unrelated to that of [Su et al. 2004].

Despite its usefulness, the 2-state Markov chain is known to have very limited ability to model long-term dependencies ([Yajnik et al. 1999]). In [Yajnik et al. 1999], a k-th order Markov model was used to capture these long term correlations. However, because of the exponential state space complexity of these models they are less attractive for on-line use. In [Salamatian and Vaton 2001], it was shown that a hidden Markov model with few states was capable of fitting those same packet loss traces from [Yajnik et al. 1999].

An issue of great relevance for modeling and predicting a time series from data is that of stationarity. Perceiving sharp variations in the statistics of interest or even deterministic phenomena such as long-term periodicity and linear trends can be some of the most challenging tasks in operations research ([Brockwell and Davis 2002]). The work in [Zhang et al. 2001] presented a comprehensive treatment of different stationarity criteria applied to measurements of real Internet end-to-end paths.

[Tao and Guerin 2004] developed a layered model for predicting end-to-end loss performance across two different time scales. Their model tries to predict the long-term loss rates and the percentage of loss bursts shorter than 3 packets, displaying small prediction error in the former while failing to perform well in the latter.

Even though these works recommend the use of models to predict packet loss performance, they do so by considering steady state measures. If network conditions are exceedingly variable on relatively short time scales, this assumption could lead to significant errors as we show in the following sections.

In [Duarte et al. 2003], the authors considered a model which aggregates the total number of lost packets in a sequence of δ attempted transmissions. The model employs restricted hidden state transitions and the prediction is applied towards selecting the proper FEC scheme for loss recovery in a VoIP tool.

In this work we persist on the idea that packet loss statistics can be reasonably well predicted if one takes care of managing the effects of non-stationarities. We support that this can be achieved by covering two important aspects of the prediction process. First, one needs a prediction mechanism with parameters capable of sensing non-stationarity quickly as it builds up. Second, but not least important, this algorithm should be coupled with a model that can structure measured data in a time scale that is appropriate to develop such sensing ability. We also strive for efficiency which is essential for any real time network control mechanism.

A more comprehensive reference on HMMs can be found in [Rabiner 1989]. Below, we present briefly the material needed for the paper. We consider models that are discrete in time, as well as in both hidden and observable states. A hidden Markov model is composed of two coupled stochastic processes. The first is a Markov chain and the other is an observation process whose distribution at any given time is fully determined by the current state of the chain.

Let $\{Y_t\}$ denote the underlying n-state Markov chain. The initial state distribution is given by the n-dimensional vector π, with $\pi_i = P(Y_1 = i)$. The state transition probabilities are controlled by the $n \times n$ matrix $\mathbf{A} = \{a_{ij}\}$, where $a_{ij} = P(Y_t = j|Y_{t-1} = i)$. The observation process $\{X_t\}$ has m states and is governed by the $n \times m$ matrix $\mathbf{B} = \{b_{ij}\}$, i.e., $b_{ij} = P(X_t = j|Y_t = i)$. We refer to the parameter set for the model as the triple $\lambda = (\pi, \mathbf{A}, \mathbf{B})$ and, given their probabilistic meanings, the constraints: $\sum_{i=1}^{n} \pi_i = 1$, $\sum_{j=1}^{n} a_{ij} = 1$ and $\sum_{j=1}^{m} b_{ij} = 1$ must hold.

A first problem in creating a new model lies in specifying the state spaces over which $\{X_t\}$ and $\{Y_t\}$ are defined. Since $\{X_t\}$ is the observation process, its states are usually determined by what is being modeled. Characterizing $\{Y_t\}$ on the other hand, may be a bit more abstract. In packet loss modeling, the hidden states may be regarded as "network states", comprising information about the packet loss statistics at a given moment. The most straightforward way to model packet loss is to represent each individual packet with a binary symbol. We consider 1 as an indicator of a loss, and 0 to mean that a packet is successfully delivered. The work in [Duarte et al. 2003] considers a different observation model consisting of 51 symbols in order to represent the total number of packet losses in a group of 50 packets. Both these approaches will be considered later in our experiments for packet loss prediction.

Consider a vector of T values for the observation process, $\mathbf{x} = [x_1, \ldots, x_T]$. Whenever there is no ambiguity, we will be using the abbreviated form $X_{i:j}$ (and accordingly $Y_{i:j}$) to denote the compound event that every variable X_t (Y_t) in the range $t = i, \ldots, j$ assumes the value x_t (y_t). In the particular case where $i = j$, we will simply be writing X_i (or equivalently Y_i). On the other hand, we will use $\mathbf{X}$ (and $\mathbf{Y}$) when the subindices span the full range $1, \ldots, T$, i.e., $X_{1:T}$ ($Y_{1:T}$). We also define the following probability measures using the notation from [Rabiner 1989]:

$$\begin{array}{rcl} \alpha_t(i) = P(X_{1:t}, Y_t = i|\lambda) & \bullet & \beta_t(i) = P(X_{t+1:T}|Y_t = i, \lambda) \\ \gamma_t(i) = P(Y_t = i|\mathbf{X}, \lambda) & \bullet & \xi_t(i,j) = P(Y_t = i, Y_{t+1} = j|\mathbf{X}, \lambda) \end{array} \quad (1)$$

Using these measures, the estimation formulae for HMMs are given as:

$$\pi_i = \gamma_1(i) \quad \bullet \quad a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad \bullet \quad b_{ij} = \frac{\sum_{t=1}^{T} \mathbb{1}\{x_t = j\}\gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \quad (2)$$

3 Packet Loss Measurements

In our experiments, we had at our disposal, an extensive set of end-to-end measurements performed between four academic sites - two of these located in Brazil and two in the United States. These measurements display a large variety of network conditions, ranging from hours with no packet loss to complete link outages. In all measurements performed, CBR traffic was generated using the tools available in the Tangram-II package [de Souza e Silva and Leão 2000].

Each traffic generation session lasted for an hour and a total of 998 sessions were performed at different periods in the years of 2001, 2002 and 2004. In any given day of experiments, the sessions were conducted at three different times, usually centered around the peak of usage in many of the links transversed, taking into account the time differences between the end points. The traffic pattern was chosen to emulate the behavior of a simplified Voice over IP (VoIP) tool, sending 50 packets with 324 bytes each per second. From our packet traces we produce a binary sequence $\{x_i\}_{i=1}^{T}$, where x_i is 1 in order to indicate a loss or 0 otherwise.

Many of the 998 collected traces exhibit packet loss statistics that are not interesting for our experimental purposes. These include statistics that are too simple to predict, such as measurements of extremely low average loss rates. We selected 194 traces whose loss rates are between 1% and 30% and whose consecutive loss periods last no more than

30 seconds. In section 7 we present quantitative results on predicting packet loss rates for these traces.

Among the traces we considered interesting to analyze quantitatively, we also selected 3 to be discussed in more detail. These traces are specially representative of network characteristics which are hard to predict. Trace 1 has a 4% loss fraction and exhibits regular spikes in its short-term loss rates with a period of 3700 packets. These heavy loss periods last approximately 1300 packets. In [Zhang et al. 2000], the authors report that routing changes can be the cause of periodic losses episodes. Our second trace has a considerable overall loss fraction of 14% with some degree of variability. Trace 3, on the other hand, displays a large number of higher short-term peaks of losses with a smaller overall loss fraction of 10%. Figure 1 shows the loss rate in the first 30 minutes of measurement of these traces. In this figure, the loss rate was evaluated at each interval of 5 seconds.

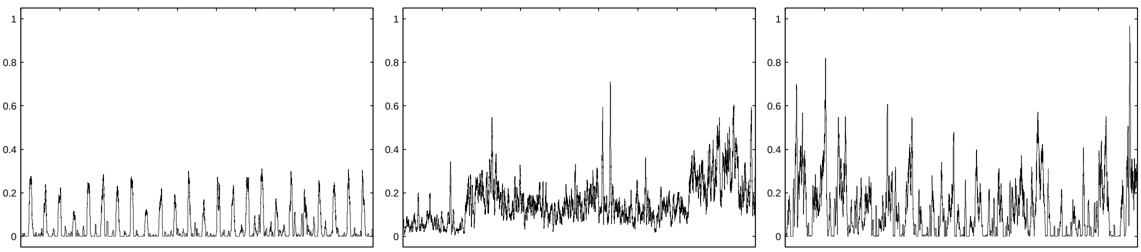

Figure 1. Loss rates of 5 second intervals in the first half hour of traces 1, 2, 3.

4 Prediction Algorithm

Steady state measures provide only a long-term average of the loss process being observed. One of our goals is to estimate a model's ability to predict loss statistics in the short-term. To achieve this goal, we calculate estimates for the loss rate from recent packet loss measures.

Estimating the short-term fraction of lost packets in the channel can be extremely important, specially if this measurement converges slowly towards steady state. Given a fixed window of f time units, the short-term loss fraction from a trace is simply the fraction of the number of loss events in this window by the number of packets transmitted in that window. Similarly to the works of [Elliott 1963] and [Su et al. 2004] we compute the exact distribution of i errors out of j transmissions. Our work is more general than that in these references since it computes this measure for any hidden Markov model while those works apply only to the Gilbert-Elliot channel. Also, our distribution for the number of losses is conditioned on the outcome of recent packet measurements. The following algorithm is an original contribution of this paper.

Let R_t^f be the random variable denoting the total number of loss events that in the next f time units beginning with the t-th observation. In other words, R_t^f is the sum of each of the f observation values, from t to $t + f - 1$:

$$R_t^f = \sum_{i=1}^{f} X_{t+i-1} \tag{3}$$

The following results can be applied to any model in which the observations represent the number of losses in a unit time, being the 0-1 model a special case. For instance, our results also apply to the hidden Markov model presented in [Duarte et al. 2003], where the observations can range from 0 to 50 losses observed in a second. Observations then lie in the range of integers from 0 to some given maximum r. As a consequence, our random variable R_t^f will span from 0 to rf.

We want to calculate the distribution of R_t^f given the h most recent past observations, $X_{t-h:t-1}$. This is the basis for our predictor of the short-term loss rate in the channel. We begin by conditioning on the hidden state of the first predicted observation:

$$P(R_t^f = j|X_{t-h:t-1}) = \sum_{\forall y_t} P(R_t^f = j|Y_t)P(Y_t|X_{t-h:t-1}) \tag{4}$$

First, we notice that this problem, as most of those related to forecasting hidden Markov models, can be broken in two steps: (a) predict the hidden state at the beginning of the future window conditioned on past observations and (b) calculate the distribution of the metric in the future conditioned on the current state.

Define $\mathbf{r}_t^{f,h}$ as the probability mass vector for R_t^f given the past history, and $\mathbf{R}^f$ as a matrix whose ij-th element is $P(R_t^f = j|Y_t = i)$. Since a hidden Markov model is a time homogeneous stochastic process, $P(R_t^f = j|Y_t = i)$ is the same measure for all t. Let $\pi_{t,h}(i) = P(Y_t = i|X_{t-h:t-1})$, i.e. $\pi_{t,h}$ is the probability vector for the hidden states at t given the past observation. We can rewrite equation (4) as:

$$\mathbf{r}_t^{f,h} = \pi_{t,h}\mathbf{R}^f \tag{5}$$

The state distribution $\pi_{t,h}$ can be easily calculated from the forward variable $\alpha_t(i)$, as defined in section 2, but measured only on the set of observations $X_{t-h:t-1}$. If we denote by α_t the vector whose i-th element is $\alpha_t(i)$, then:

$$\pi_{t,h} = \frac{\alpha_{t-1}\mathbf{A}}{P(X_{t-h:t-1})} \tag{6}$$

In order to devise a recursion that evaluates $\mathbf{R}^f$, one should note that $P(R_t^1 = j|Y_t = i) = P(X_t = j|Y_t = i) = b_{ij}$. Hence, matrix $\mathbf{R}^1$ is the observation matrix $\mathbf{B}$.

Conditioning on the value of the subsequent observation, X_t, we can write each element of $\mathbf{R}^f$ as:

$$P(R_t^f = j|Y_t = i) = \sum_{\forall x_t} \left[\sum_{\forall y_{t+1}} P(R_{t+1}^{f-1} = j - x_t|Y_{t+1})a_{iy_{t+1}} \right] b_{ix_t} \tag{7}$$

We can rewrite (7) in matrix form as:

$$\mathbf{R}^k = \begin{cases} \mathbf{B} & , \quad k = 1 \\ \sum_{\forall x_t} \mathbf{B}(x_t)\mathbf{A}\mathbf{R}^{k-1}\mathbf{I}_f(x_t) & , \quad 2 \le k \le f \end{cases} \tag{8}$$

Where $\mathbf{B}(x_t) = \mathrm{diag}\{b_{ix_t}\}$ and $\mathbf{I}_f(x_t)$ is a $(1+rf-r)\times(1+rf)$ matrix identical to an $(1+rf-r)\times(1+rf-r)$ identity matrix shifted x_t columns to the right and with zeros on all the remaining elements. The steps in our algorithm are:

$$\begin{array}{rl} \text{Initialization} & \left\{ \begin{array}{l} \pi_{t,h} \leftarrow \alpha_{t-1}\mathbf{A} \\ \mathbf{R}^1 \leftarrow \mathbf{B} \end{array} \right. \\ \text{Main loop} & \left\{ \begin{array}{l} \text{for } 2 \le k \le f \text{ do:} \\ \quad \mathbf{R}^k \leftarrow \sum_{\forall x_t} \mathbf{B}(x_t)\mathbf{A}\mathbf{R}^{k-1}\mathbf{I}_k(x_t) \end{array} \right. \\ \text{Result} & \left\{ \mathbf{r}_t^{f,h} \leftarrow \pi_{t,h}\mathbf{R}^f \right. \end{array} \tag{9}$$

4.1 Adaptive Prediction Mechanism

We now develop an adaptive prediction mechanism for training the HMM and evaluating the measure predictor. Figure 2 provides a general picture of our methodology in two layers. In the *model training layer*, model parameters are periodically re-estimated every τ time units. In each training, only samples from the latest T time slots are used for the parameter estimation procedure. Every training epoch is also divided in prediction intervals of length ψ, as shown in the *measure prediction layer*. Each individual prediction can be conditioned on the packet samples from the h most recent time slots. We also have a parameter f that specifies the size of the prediction window, i.e., the maximum number of packets that can be lost.

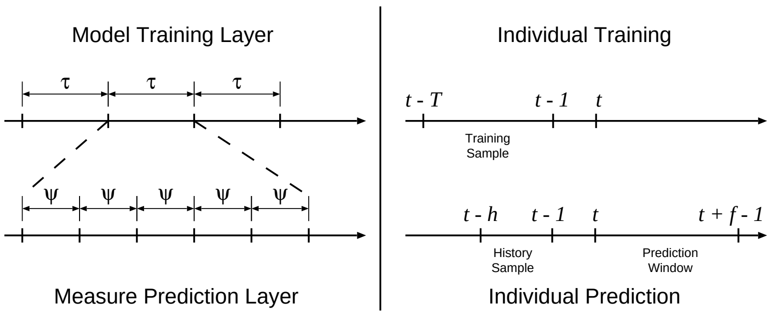

Figure 2. Prediction mechanism layers

Through experimentation, we have found that each of these parameters can have different impacts on the quality of prediction. The values of T and h, for instance, play important roles in perceiving the effects of recent changes in channel statistics. If either one is set too high, predictions are much smoother, basically reflecting the steady state measures. Values that are too low, on the other hand, will fail to include enough information to allow the model to correctly estimate its parameters or perform accurate prediction. Clearly, in a real scenario, one would like to have the values of τ and ψ as large as possible to minimize application overhead. Nevertheless, there is a trade-off between this overhead and the accuracy of prediction.

The packet loss information used for conditioning the prediction statistics is, in practice, not available to the sender immediately after they occur. Since the sender must wait a round-trip-time until prediction can be performed, some part of the predicted statistics will be useless in taking control decisions. Because of this, in the loss rate prediction mechanism, the value of f should not be chosen too small in relation to the estimated RTT. On the other hand, it is easy to see that if one makes $f \to \infty$, the predicted loss rate will be independent of the history given by h and converge to the steady state loss probability. In our experiments we have tried a number of different variations for all of these parameters. Since an extensive presentation of these comparisons is out the scope of this paper, in the next section we present results based on parameter values that we have found to work well in practice.

5 Experimental Results

We now compare the accuracy of two different models to predict the short term loss rate using the method presented in the previous section. In this section, our analysis is focused on the three traces selected in section 3 which we consider hard to predict. In section 7, we present quantitative results for the larger set of 194 traces.

The first model we consider is a hidden Markov model with binary observations and 10 hidden states, which models individual packet loss events, and for such we refer to it as a Packet-HMM. The second model is another HMM, also with 10 hidden states, although its observation measure is the total number of losses in a sequence of 50 packets, which for our traffic specification corresponds to an interval of 1 second. We denote this last model the Aggregate-HMM.

For all models, we apply the algorithm from section 4 in order to calculate the distribution for the number of losses in the next $f = 5$ seconds, given the outcomes of the packet losses in the last $h = 10$ seconds. These prediction estimates are updated every $\psi = 5$ seconds. Model parameters are re-estimated every $\tau = 3$ minutes using the information from the latest $T = 3$ minutes.

Once the distribution for the loss rate is evaluated for a model, we use its expected value as a predictor statistic and take its absolute difference to the real measure obtained from the trace as a *prediction distance*. We consider as an accurate prediction, one in which such distances are as small as possible most of the time. Thus, we compare the expected value obtained from the algorithm to the actual number of losses observed in the corresponding segment of the trace. We also experimented, for each model, using the long-term loss probability given the current parameters as an estimator of the packet loss rate in the next f time slots. This predictor does not make use of the algorithm in (9).

We present the performance of each model in predicting the loss rate using two different metrics. In the first one, we simulate a *contest* where at any given prediction instant, each of the competing models uses as its score the distance measure defined in the previous paragraph. The model with the lowest score is the winner in that round. We then evaluate, for the entire trace, which model has won the most rounds.

The second measure of accuracy we consider is the *mean squared error*, that is, the average of the squares of the *prediction distances*. The MSE can be misleading for comparisons since a single bad prediction will cause the entire measure to look bad. We recognize that there is no single measure that is good enough to reflect the accuracy of a predictor. For this reason, we do not compare the MSE for each model, but instead we display its value only for the one that wins the prediction distance competition that we defined above.

Trace	Packet-HMM	Aggregate-HMM	MSE of Best Model
1	47.10%	52.90%	0.00484
2	58.54%	41.46%	0.00654
3	45.97%	54.03%	0.01411

Table 1. Fraction of time each model provides the closest prediction.

For the traces discussed in section 3 the results of our metrics are displayed in table 1 for the transient predictors of loss rate in each model. Except for the second trace, the Packet-HMM was not as good as the Aggregate-HMM. This happens because in the Packet-HMM, the channel statistics may change after each packet transmitted. As a consequence, steady state will be reached much faster than that for the Aggregate-HMM, in which hidden state transitions occur at a longer time scale.

Figure 3 illustrates these characteristics in more detail. It shows the prediction performance of each model in the first 30 minutes of trace 3. In the left plot, it can be noticed that the Packet-HMM was not capable of reproducing the variations of the loss rate in the short-time scales. In fact the only variations that are visually evident occur in large steps, every $\tau = 3$ minutes, when the model parameters are re-estimated. The Aggregate-HMM, on the right, better predicts the intensity of variations that occur between the training epochs.

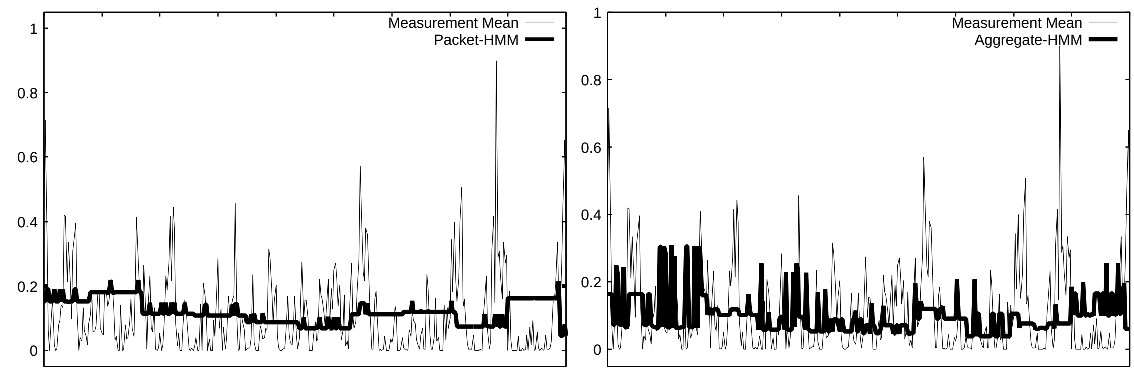

Figure 3. Prediction results in trace 3.

We also found that, in all traces, the transient prediction of loss rates provided by our algorithm is consistently better than the steady state alternative, since the latter simply ignores any variations in recent statistics and predicts the same outcome for an entire training epoch.

To emphasize the impact of the slow converge towards steady state for predicting loss rates, we compare the prediction done in two different ways: (a) using the Aggregate-HMM that had the best performance for trace 3 in figure 3 and; (b) using the steady state results of the same model for each prediction interval. Figure 4 shows the results for the steady state predictor. One can observe that, as expected, the algorithm fails to predict the trace variability during each prediction interval. This contrasts with the plot on the right of figure 3 where transient measures were used and so the algorithm was able to track reasonably close the variability of the loss rate during the interval of length ψ.

Finally, we have studied the sensitivity of the prediction with parameter τ. For

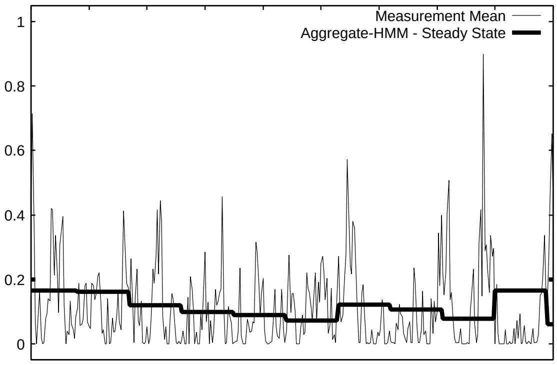

Figure 4. Results of using the steady state measure as a predictor in trace 3.

instance, if τ is set to 1 minute, the Packet-HMM will be able to update its predictions as fast as the Aggregate-HMM has done for $\tau = 3$ minutes. However, while the aggregate model is adapting its forecasts in these three minutes based only on the prediction algorithm, the Packet-HMM's improvement in performance is just due to the fact that its parameters are now being updated more frequently than before. If, on the other hand, τ is increased to 5 minutes, then the Aggregate-HMM becomes less accurate than for $\tau = 3$ minutes, whenever a large change in the loss rate occurs.

6 Proposed Model

In this section we propose a hierarchical hidden Markov model that has lower computational complexity than those described in previous sections. The computational savings are based on the premise that the statistics changing in short time scales can be reasonably well predicted with simple models. We also discuss how the prediction algorithm from section 4 can be adapted for the proposed model.

Suppose that transitions between hidden states occur only every S observations. Another way to think of this process is to assume that, once it enters a hidden state, it emits a *batch* of S packet transmission outcomes. For this reason, we refer to this as *batch-observations model*. Clearly, the case of $S = 1$ is equivalent to an "ordinary" hidden Markov model.

This process can be modeled by a HMM in which a state can emit one of the 2^S possible observation symbols, i. e. one for each possible sample path for S packets. However, this model would be unmanageable even for moderate values of S. In our approach, we restrict the distribution of observations within a batch by assuming that it is generated by a simplified Gilbert model, i.e., a 2-state Markov chain. The reasoning behind our model is that short-term correlations could be captured by a simple process, while the dynamics at larger time scales would be governed by the hidden Markov chain. Computational savings are achieved by considering a batch of S measurements as a single observation, and computing the joint probability of the entire batch from the distribution of the generating process in each state.

We consider that packet measurements are segmented in sets of size S. More specifically, let the measurement symbols x_t denote a vector of measures $[x_{t,1}, \ldots, x_{t,S}]$ representing the outcome for each of the individual packets grouped in the t-th batch. Accordingly we redefine observation variables X_t as vectors of variables $[X_{t,1}, \ldots, X_{t,S}]$.

For each hidden state we have the parameters for the 2-state Markov chain. Namely:

$$r_i = P(X_{t,1} = 1 | Y_t = i) \tag{10a}$$
$$p_i = P(X_{t,s} = 1 | X_{t,s-1} = 0,\, Y_t = i), \quad s > 1 \tag{10b}$$
$$q_i = P(X_{t,s} = 0 | X_{t,s-1} = 1,\, Y_t = i), \quad s > 1 \tag{10c}$$

We refer to the model as the tuple $\lambda = (\pi, \mathbf{A}, \mathbf{r}, \mathbf{p}, \mathbf{q})$, where $\mathbf{r}, \mathbf{p}, \mathbf{q}$ are vectors containing the respective parameters r_i, p_i, q_i, for each state i.

The first advantage of our model is the lower computational complexity than the previous models presented. This is true since, in order to evaluate the likelihood of a sample, one does not need to record the individual packet measurements. It is enough to keep track of the *sufficient statistics*[1] denoted, in each batch of measures x_t, as:

$$x_{t,1} = \text{outcome of the first packet in } x_t \tag{11a}$$
$$S^{ij}_{x_t} = \text{number of transitions from i to j in } x_t, \quad i, j \in \{0, 1\} \tag{11b}$$

Given an instance of x_t, we are be interested in computing the probability that $X_t = x_t$, given the hidden state y_t in the t-th batch. Using the statistics defined above, we have:

$$b_{y_t,x_t} = \begin{cases} r_{y_t}(p_{y_t})^{S^{01}_{x_t}}(1-p_{y_t})^{S^{00}_{x_t}}(q_{y_t})^{S^{10}_{x_t}}(1-q_{y_t})^{S^{11}_{x_t}} & , \text{ if } x_{t,1} = 1 \\ (1-r_{y_t})(p_{y_t})^{S^{01}_{x_t}}(1-p_{y_t})^{S^{00}_{x_t}}(q_{y_t})^{S^{10}_{x_t}}(1-q_{y_t})^{S^{11}_{x_t}} & , \text{ if } x_{t,1} = 0 \end{cases} \tag{12}$$

We present a theorem that allows us to efficiently compute the parameters given by equations (10).

Theorem 1. *If restrictions are added to either one of the subsets of parameters* π, $\mathbf{A}$ *or* $\mathbf{B}$, *in addition those mentioned in section 2, the re-estimation formulae will change only for those variables upon which the new restrictions apply, as long as the new restrictions do not involve parameters in different subsets.*

We omit the proof of Theorem 1 for space restrictions. In our methodology we develop a new restriction on the observation parameters $\mathbf{B}$. The estimation formulae for π and $\mathbf{A}$ remain unchanged nevertheless.

Using the results of Theorem 1 and applying the restrictions in (12) on the observations parameters b_{ij}, it is possible to obtain the new re-estimation formulae:

$$r_i = \frac{\sum_{t=1}^{T} \mathbf{1}\{x_{t,1}=1\}\gamma_t(i)}{\sum_{t=1}^{T}\gamma_t(i)} \quad \bullet \quad p_i = \frac{\sum_{t=1}^{T} S^{01}_{x_t}\gamma_t(i)}{\sum_{t=1}^{T}(S^{01}_{x_t}+S^{00}_{x_t})\gamma_t(i)} \quad \bullet \quad q_i = \frac{\sum_{t=1}^{T} S^{10}_{x_t}\gamma_t(i)}{\sum_{t=1}^{T}(S^{10}_{x_t}+S^{11}_{x_t})\gamma_t(i)} \tag{13}$$

The computational advantage of our model is evident from equations (13), since each training iteration depends only on the statistics defined in (11). Each training iteration is therefore faster by a factor of S. Since the measurements used in training are usually done at the receiving side of the transmission and need to be sent back to the transmitter, there is also a smaller overhead in application payload due to these out-of-band data.

[1] A statistics $\Gamma(x)$ is a *sufficient statistic* for θ if the distribution of the sample X given the value of $\Gamma(x)$ does not depends on θ.

We now show how to calculate the distribution of R_t^f in our batch observation models. We restrict our analysis to the simpler case where both t and f are multiples of the batch size S, i.e., $t = t'S$ and $f = f'S$ for some integers t' and f'. The general case is notationally intensive and it is omitted for conciseness.

Separating the number of losses from each batch of observations, equation (3) can be rewritten as:

$$R_t^f = \sum_{i=0}^{f'-1} R_{t+iS}^S = \sum_{i=0}^{f'-1} \sum_{j=1}^{S} X_{t+iS+j} \tag{14}$$

Note that, conditioning on the same hidden state, each of the terms R_{t+iS}^S is independent and identically distributed. We can then create an aggregate hidden Markov model that counts the number of losses in each batch, similarly to the one we considered in the experiments of the previous section.

Once matrix $\mathbf{B}$ is evaluated for the aggregate model, one can merely apply the algorithm given by (9) in order to obtain the distribution for R_t^f. Each row $\mathbf{b}_i$ of the aggregate observation matrix $\mathbf{B}$ is defined as the probability distribution for the number of losses in a batch of size S, generated by the 2-state Markov chain model contained in the corresponding state i. It is interesting to notice that, in order to evaluate $\mathbf{b}_i$ in this case, one can apply the algorithm given by (9) with some minor simplifications.

7 Additional Experimental Results

We report the performance of our prediction algorithm when used together with the batch-observations model. The model used for the experiments has 10 hidden states and $S = 50$ packet outcomes between state transitions. This is equivalent to 1 second of packet transmissions for our packet traces that emulate a voice application. We refer to our new model as the Batch-HMM.

In order to quantify the advantage of the Batch-HMM over the Packet and Aggregate HMMs as a predictor we consider another metric - the sample cross-correlation between predicted and real loss rates - which should be as large as possible to indicate a good prediction. Namely, if a predictor is coherent with the fluctuations in the loss rate, its predictions should correlate well with the real measurements.

Figure 5 shows the sample correlations obtained for the prediction of each model against our 194 traces. Each of the three curves corresponds to one of the models we are comparing. The traces are arbitrarily sorted in the x axis so as to make the curve corresponding to the Batch-HMM non-decreasing.

It is evident from the plot in Figure 5 that most of time, the correlations from the Packet-HMM and the Aggregate-HMM fall below the reference curve from the Batch-HMM. In fact, the Batch-HMM provides the highest correlation in 49% of the traces, while the Packet and Aggregate models do so, respectively, in 40% and 11% of the traces. Also, in the traces where the Batch-HMM failed to provide the highest correlation, the difference to the best model was never larger to 0.18.

In table 2, we compare the performances of the Batch-HMM with that of the best models for each trace considered in the experiments of section 5 (shown in table 1). As

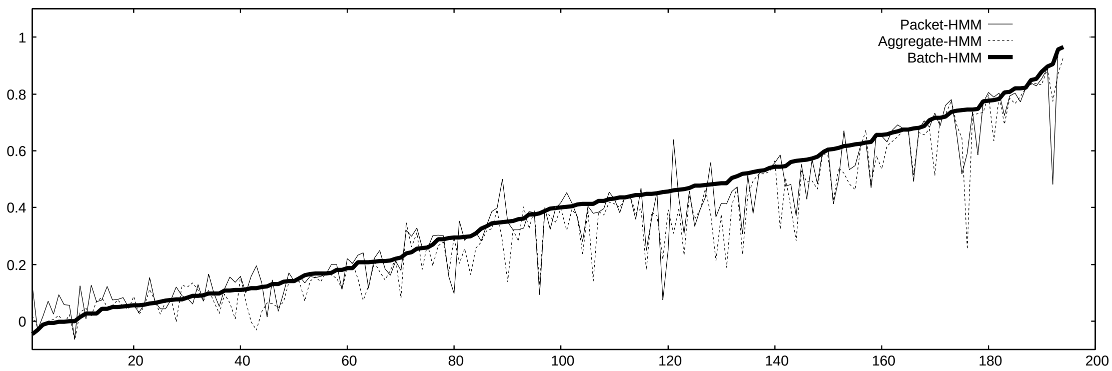

Figure 5. Correlations between prediction and real loss rates in the 194 traces.

Trace	Wins against Packet-HMM	Wins against Aggregate-HMM
1	57.43%	51.20%
2	56.46%	61.37%
3	58.42%	57.28%
4	66.34%	57.28%

Table 2. Fractions of time the Batch-HMM provides the closest prediction against the Packet and Aggregate models.

in that section, the comparison measure is the fraction of time a model provided the best prediction among all the considered models.

Figure 6 shows the performance of the Batch-HMM in predicting the first 30 minutes of data in trace 3. If we compare Figure 6 against 3, we observe that our model better adapts the prediction of the loss rate to the large variations that occur in the trace than the other models considered. Similarly to the Aggregate-HMM, the Batch model converges to steady state at a slower pace than the Packet-HMM. On the other hand, the Batch-HMM has fewer parameters values to be estimated than the Aggregate-HMM and therefore these can be estimated more efficiently and potentially more accurately.

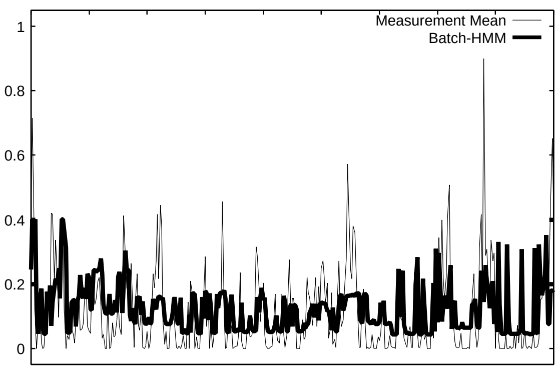

Figure 6. Prediction results for the Batch-HMM in trace 3.

As observed in section 3, the loss rate of trace 1 contains periodic spikes at every 74 seconds. Figure 7 shows a 10 minutes zoom over the first trace plot of Figure 1, together with the prediction made by our algorithm when using the Aggregate-HMM (on the left) and Batch-HMM (on the right). The figure clearly indicates that our model better adapts to the temporary loss rate sharp changes. As an example application, this predicting capability can be of great value for a VoIP tool that selects dynamically, in real-time, a FEC scheme to mitigate the harmful effects of these loss rates spikes in the perceived voice quality.

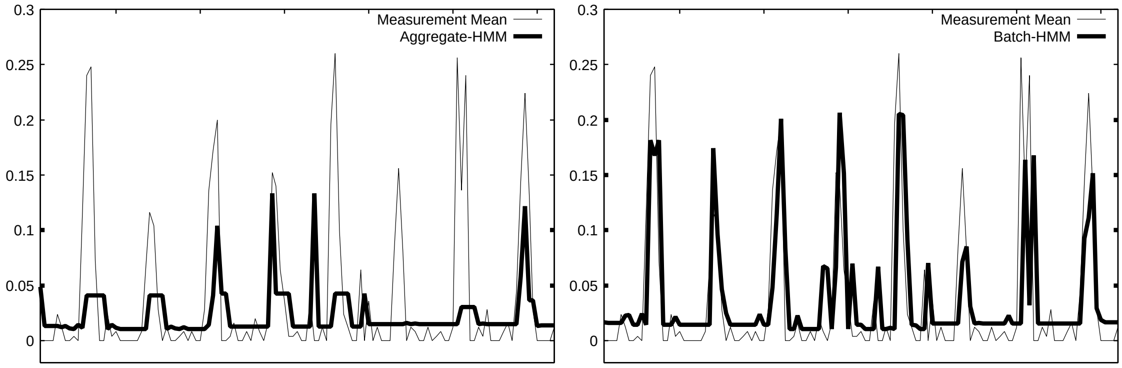

Figure 7. Prediction results for the Aggregate and Batch HMMs in trace 1.

Figure 8 shows the predicting performance of our algorithm when applied to trace 2 using the Packet-HMM and the Batch-HMM. The figure shows that the algorithm better predicts short term loss rate variations when the Batch-HMM is used. Nevertheless, even the Batch-HMM algorithm has difficulties to perceive very sharp changes. This is evident by the "step-like" shape of the prediction trace. We attribute the prediction errors to the fact that trace 2, as reported in section 3, has a very small sample autocorrelation in all time lags when compared to those of the other traces. Evidently, any prediction is limited by the amount of temporal correlations in a trace. The smaller the temporal correlations the harder is the prediction task.

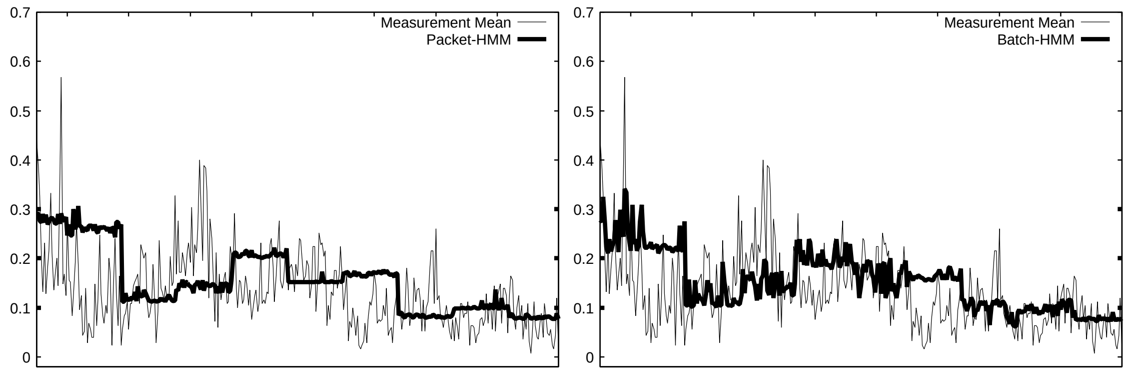

Figure 8. Prediction results for the Packet and Batch HMMs in trace 2.

8 Computational Costs

We evaluate the the efficiency of the Packet, Aggregate and Batch HMMs when used in conjunction with the methodology described in section 4.1. We consider models with N hidden states, that can aggregate r packets per time slot, evaluating the distribution for the number of losses in the next f time slots.

Concerning the model training step, each iteration of the training procedure of HMMs has its complexity dominated by the forward-backward recursions. For a training sample of size T, the Packet-HMM performs a number of computations in the order of N^2T. In both Aggregate and Batch models, packet loss information from the training sample is described in the form of sufficient statistics measured over portions of the sample of size S. Therefore, the complexity of the forward-backward steps is reduced by a factor of S in these models.

As for the measure prediction algorithm in (9), the number of operations that need to be taken to evaluate the matrix $\mathbf{R}^f$ is approximately given by:

$$C(N,r,f) = N^2\left[\frac{(r^2+r)(f-1)f}{2} + \frac{r(f-2)(f+1)}{2} + 2(f-1)\right] \tag{15}$$

For the experiments we have reported in sections 5 and 7, the Aggregate and Batch models execute 3 times less operations than the Packet-HMM. This is a significant improvement for a real time predictor.

9 Conclusions and Future Work

In this paper we devise a novel online prediction algorithm for packet losses. We also proposed a hierarchical loss model aimed at capturing short-term variations. This model, when used with the prediction algorithm is sufficiently accurate to predict error rates within a reasonable time frame in the future. The parameter estimation procedure for the proposed model uses new equations that we developed for this work. We showed that these equations lead to significant computational savings for the prediction algorithm, in contrast with the same algorithm using other models in the literature.

We evaluated the prediction algorithm using three different loss models and over a number of traces collected in the Internet. We also discussed in more detail the results from three of these traces. These correspond to loss processes that are hard to predict as argued in the paper. Our results have shown that our proposed model is significantly more efficient in terms of parameter estimation and also outperforms the others in most cases. It is therefore the model of choice for our algorithm.

We also conclude that the short-term loss rate can not be well approximated by the steady state loss probability, since this can result in poor predictions. In addition, our findings also show that the short-term predictions of the algorithm when the Packet-HMM is employed do not capture high variations in the loss rate. This is true because these models reach steady state within a short interval. We show that both the Aggregate-HMM and the Batch-HMM are more adequate to model these transient effects than the other two models.

It is clear that the accuracy of prediction is limited by the amount of temporal correlations in the packet loss process. Traces with relatively small time dependencies reduce our model's ability to predict short-term packet loss rates.

The prediction algorithm seems to be of value when applied to a real time streaming applications such as VoIP and video-conferencing, or even other applications that can benefit from foreseeing error rates. We have plans to incorporate the algorithm as part of an adaptive control protocol in the future.

10 Acknowledgements

We would like to thank Flavio P. Duarte for providing us with an implementation of the hidden Markov model training algorithms. David S. Boechat and Hugo H. C. Sato have also been very helpful in the implementation of other tools for the experiments that have led to this paper. We are particularly in debt with David, who helped us in generating the results that we report here and also in revising earlier versions of this article.

References

Bolot, J. C., Parisis, S. F., and Towsley, D. F. (1999). Adaptive FEC-based error control for Internet telephony. In *Proceedings of the IEEE INFOCOM*, pages 1453–1460.

Brockwell, P. J. and Davis, R. A. (2002). *Introduction to Time Series and Forecasting*. Springer.

de Souza e Silva, E. and Leão, R. M. M. (2000). The TANGRAM-II Environment. In *Lecture Notes In Computer Science - Volume 1786*, pages 366–369, London, UK. Springer-Verlag.

Duarte, F. P., de Souza e Silva, E. A., and Towsley, D. F. (2003). An adaptive FEC algorithm using hidden Markov chains. *Performance Evaluation Review*, 31(2):11–13.

Elliott, E. O. (1963). Estimates of error rates for codes on burst-noise channels. *Bell Systems Technical Journal*, 42:1977–1997.

Elliott, E. O. (1965). A model of the switched telephone network for data communications. *Bell Systems Technical Journal*, 44:89–109.

Karol, M., Krishnan, P., and Li, J. J. (2004). Rapid fault detection and recovery for IP telephony. In *Proceedings of IEEE INFOCOM*.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285.

Salamatian, K. and Vaton, S. (2001). Hidden Markov modeling for network communication channels. In *Proceedings of the ACM SIGMETRICS*, pages 92–101.

Su, Y. C., Yang, C. S., and Lee, C. W. (2004). The analysis of packet loss prediction for Gilbert-model with loss rate uplink. *Information Processing Letters*, 90:155–159.

Tao, S. and Guerin, R. (2004). On-line estimation of Internet path performance: an application perspective. In *Proceedings of IEEE INFOCOM*.

Tao, S., Xu, K., Estepa, A., Fei, T., Gao, L., Guerin, R., Kurose, J. F., Towsley, D. F., and Zhang, Z. (2005). Improving VoIP quality through path switching. In *Proceedings of IEEE INFOCOM*.

Yajnik, M., Moon, S. B., Kurose, J. F., and Towsley, D. F. (1999). Measurement and modeling of the temporal dependence in packet loss. In *Proceedings of the IEEE INFOCOM*, pages 345–352.

Zhang, Y., Duffield, N. G., Paxson, V., and Shenker, S. (2001). On the constancy of Internet path properties. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 197–211.

Zhang, Y., Paxson, V., and Shenker, S. (2000). The stationarity of Internet path properties: routing, loss, and throughput. Technical report, ACIRI.