A HIDDEN MARKOV MODEL FOR INTERNET CHANNELS

Pierluigi Salvo Rossi (1), *Gianmarco Romano* (2), *Francesco Palmieri* (2), *Giulio Iannello* (1,3)

(1) Dipartimento di Informatica e Sistemistica - Università di Napoli "Federico II"
(2) Dipartimento di Ingegneria dell'Informazione - Seconda Università di Napoli
(3) Università Campus Bio-Medico di Roma

ABSTRACT

Performance of real-time applications on network communication channels are strongly related to losses and temporal delays. Several studies showed that these network features may be correlated and present a certain degree of memory such as bursty losses and delays. The memory and the statistical dependence between losses and temporal delays suggest that the channel may be well modelled by a Hidden Markov Model (HMM) with appropriate hidden variables that capture the current state of the network. In this paper we propose an HMM that, trained with a modified version of the EM-algorithm, shows excellent performance in modelling typical channel behaviors in a set of real packet links.

1. INTRODUCTION

Gilbert and Elliott works [1][2] on modelling burst-error channels for bit-transmission showed how a simple 2-states Hidden Markov Model (HMM) was effective in characterizing some real communication channels. As in the case of bit-transmission channels, end-to-end packet channels show burst-loss behavior. Jiang and Schulzrinne [10] investigated lossy behavior of packet channels finding that a Markov model is not able to describe appropriately the inter-loss behavior of channels. They also found that delays manifest temporal dependency, i.e. they should not be assumed to be a memoryless phenomenon. Salamatian and Vaton [11] found that an HMM trained with experimental data seems to capture channel loss behavior. Liu, Matta and Crovella [12] used an HMM-based loss-delay modelling in the contest of TCP traffic in order to infer loss nature in hybrid wired/wireless environments. They found that such a kind of modelling can be used to control TCP congestion avoidance mechanism. Similar works have been done by Zorzi [7] on wireless fading links.

These works suggest that a Bayesian state-conditioned model may be effective in capturing the dynamic behavior of losses and delays on end-to-end packet channels. The definition of a model capturing jointly losses and delays is highly desirable for designing and evaluating coding strategies, such as Multiple Description Coding (MDC), Forward Error Correction (FEC), Error Concealment (EC). Furthermore, the possibility of learning on-line the model parameters opens the way to design efficient content adaptation services.

In this paper we propose a comprehensive model that jointly describes losses and delays and propose a version of the EM algorithm for learning the model parameters. A set of simulations based on measurements obtained on real packet links confirms how effective the model can be.

This work has been carried out partially under the financial support of the Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) in the framework of the FIRB Project "Middleware for advanced services over large-scale, wired-wireless distributed systems (WEB-MINDS)".

Fig. 1. End-to-end packet channel.

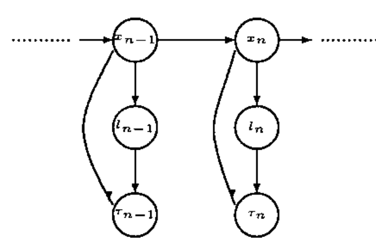


Fig. 2. The Bayesian model for packet channel.

2. THE MODEL

Fig. 1 shows our reference model with a periodic source traffic with inter-departure period T and fixed packet size of N_b bits. The system data rate is $R = N_b/T$ $bits/s$. The network randomly cancels and delays packets according to current congestion. Transmitted packets are numbered, $n = 1, 2, \ldots$; t_n and τ_n are the arrival time and the accumulated delay of the *n-th* packet respectively, i.e. $\tau_n = t_n - nT$.

The presence of memory in the phenomena suggested to introduce a hidden state variable that stochastically influences losses and delays. The state variable is hidden because our knowledge about it can only be inferred from observation of losses and delays. Let us denote x_n the state of the link at time step n, with $x_n \in \{s_1, s_2, \ldots, s_N\}$, s_i being the *i-th* state, and $l_n \in \{v_1, v_2\}$ a binary variable where v_1 and v_2 correspond respectively to absence or presence of a loss. Our reference Bayesian model is shown in Fig. 2 where the arrows represent statistical dependence among variables. The model can be reduced to an HMM [14], shown in Fig. 3, with a hidden variable x_n and an observable vari-

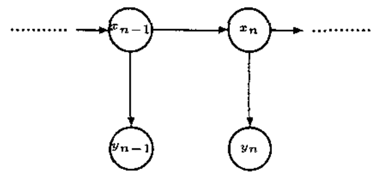


Fig. 3. Hidden Markov Model.

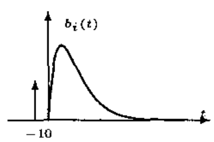


Fig. 4. An example of the conditional pdf $b_i(t)$ for the hybrid variable y_n.

able y_n that represents jointly losses and delays as

$$y_n = \begin{cases} \tau_n & if \quad l_n = v_1 \\ -1 & if \quad l_n = v_2 \end{cases} . \qquad (1)$$

$\Lambda = \{\mathbf{A}, \mathbf{p}, f_1(\tau), f_2(\tau), \ldots, f_N(\tau)\}$ is the set of parameters characterizing the model, where: $\mathbf{A} = [a_{ij}]_{i,j=1}^N$is the state transition matrix, $\mathbf{p} = [p_i]_{i=1}^N$is the loss probability vector, and $\{f_i\}_{i=1}^N$are the delay conditional pdf,i.e.

$$a_{ij} = Pr(x_{n+1} = s_j | x_n = s_i) \;_{i,j \in \{1,2,\ldots,N\}} , \qquad (2)$$

$$\begin{cases} p_i = Pr(l_n = v_1 | x_n = s_i) \\ 1 - p_i = Pr(l_n = v_2 | x_n = s_i) \end{cases} \;_{i \in \{1,2,\ldots,N\}} , \qquad (3)$$

$$.Pr(\tau_n > t | x_n = s_i, l_n = v_1) = \int_t^{+\infty} f_i(\tau) d\tau . \qquad (4)$$

Summarizing:

- x_n is a discrete random variable whose dynamic behavior is governed by the transition matrix $\mathbf{A}$;
- y_n is a hybrid random variable that, given $\{x_n = s_i\}$, is characterized by the pdf,

$$b_i(t) = p_i f_i(t) + (1 - p_i)\delta(t + 1) . \qquad (5)$$

The hybrid variable y_n is obtained as a mixture of two components (one continuous, the other discrete), where there is a probability mass concentrated in -1 to model losses while the continuous component describes network delays behavior in the absence of losses, see Fig. 4.

If $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$ is the stationary state probability distribution, i.e.

$$\pi_i = \lim_{n \to \infty} \{Pr(x_n = s_i)\} \;_{i \in \{1,2,\ldots,N\}} , \qquad (6)$$

the average loss probability and the average delay of the model are:

$$P_{loss} = \sum_{i=1}^N \pi_i (1 - p_i) , \; \overline{d} = \sum_{i=1}^N \pi_i \int_0^{+\infty} t f_i(t) dt . \qquad (7)$$

3. LEARNING THE MODEL PARAMETERS

The Expectation-Maximization (EM) algorithm [8] is an optimization procedure that allows learning of a new set of parameters for a stochastic model according to improvements of the likelihood of a given sequence of observable variables. For structures like HMM of Fig. 3 this optimization technique reduces to the Forward-Backward algorithm [3][4][5] studied for discrete and continuous observable variables with a broad class of allowed conditional pdf's. More specifically, given a sequence of observable variables $\mathbf{y} = (y_1, y_2, \ldots, y_K)$ referred to as the *training sequence*, we want to find the set of parameters such that the likelihood $L(\mathbf{y}; \Lambda) = Pr(\mathbf{y}|\Lambda)$ of the training sequence is maximum.

The Forward-Backward algorithm is an iterative procedure looking for a local maximum of the likelihood function which typically depends on the starting point Λ. When necessary, repeated starts with different initial conditions provide the global solution. It is based on the following equations:

$$\hat{a}_{ij} = \frac{\sum_{k=1}^{K-1} \alpha_k(i) a_{ij} b_j(y_{k+1}) \beta_{k+1}(j)}{\sum_{k=1}^{K-1} \alpha_k(i) \beta_k(i)} \;_{i,j \in \{1,2,\ldots,N\}} , \qquad (8)$$

$$\hat{p}_i = \frac{\sum_{k=1}^{K} \rho_k(i) \beta_k(i)}{\sum_{k=1}^{K-1} \alpha_k(i) \beta_k(i)} \;_{i \in \{1,2,\ldots,N\}} , \qquad (9)$$

$$\hat{\mu}_i = \frac{\sum_{k=1}^{K} \rho_k(i) \beta_k(i) y_k}{\sum_{k=1}^{K-1} \rho_k(i) \beta_k(i)} \;_{i \in \{1,2,\ldots,N\}} , \qquad (10)$$

$$\hat{\sigma}_i^2 = \frac{\sum_{k=1}^{K} \rho_k(i) \beta_k(i) (y_k - \mu_i)^2}{\sum_{k=1}^{K-1} \rho_k(i) \beta_k(i)} \;_{i \in \{1,2,\ldots,N\}} , \qquad (11)$$

where

$$\alpha_k(j) = \sum_{i=1}^N \alpha_{k-1}(i) a_{ij} b_j(y_k) \;_{\substack{k \in \{2,3,\ldots,K\} \\ j \in \{1,2,\ldots,N\}}} , \qquad (12)$$

$$\beta_k(i) = \sum_{j=1}^N a_{ij} b_j(y_{k+1}) \beta_{k+1}(j) \;_{\substack{k \in \{1,2,\ldots,K-1\} \\ i \in \{1,2,\ldots,N\}}} , \qquad (13)$$

are the forward and backward partial likelihood, and where

$$\rho_k(j) = \sum_{i=1}^N \alpha_{k-1}(i) a_{ij} p_j \left. \frac{\partial b_j(t)}{\partial p_j} \right|_{t = y_k} \;_{\substack{k \in \{2,3,\ldots,K\} \\ j \in \{1,2,\ldots,N\}}} . \qquad (14)$$

The problem of the Dirac-impulse in the conditional pdf, Eq.(5), is avoided considering a modified function

$$\tilde{b}_i(t) = p_i f_i(t) + (1 - p_i) g(t) , \qquad (15)$$

where $g(t)$ is any pdf such that

$$g(t) = 0 , \; \forall \, t \geq 0, \qquad (16)$$

to avoid overlapping supports between $f_i(t)$ and $g(t)$. Obviously, while the set $\{f_i(t)\}_{i=1}^N$ will be adjusted by the iterative procedure, $g(t)$ will remain unchanged, as only its area is relevant. This means that losses, in the algorithm can be randomized according to $g(t)$.

Our choice of conditional pdf's for modelling delays is a classical Gamma distribution, as suggested by several works [6][9],

$$f_i(t) = \frac{(t/\vartheta_i)^{\gamma_i - 1} e^{-(t/\vartheta_i)}}{\vartheta_i \Gamma(\gamma_i)} u(t) . \qquad (17)$$

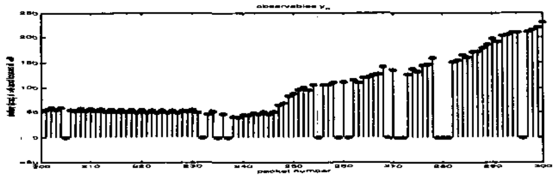


Fig. 5. Portion of a measured trace on a real network.

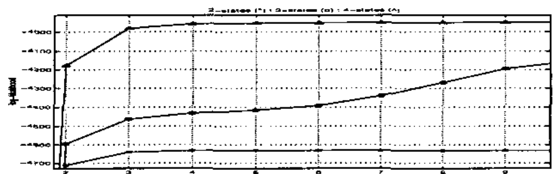


Fig. 6. Example of log-likelihood trend in the learning procedure, 2−state model (∗), 3−state model (∘), 4−state model (△).

while losses are simply randomized according to a narrow uniform distribution around −1, i.e.

$$g(t) = \begin{cases} \frac{1}{2\Delta} & t \in [-1-\Delta, -1+\Delta] \\ 0 & otherwise \end{cases} , \quad (18)$$

where Δ is an arbitrary small number.

4. EXPERIMENTAL RESULTS

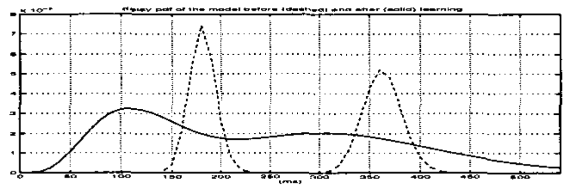
Measures of losses and delays have been performed on some real networks using the software Internet Traffic Generator (ITG) [13]. ITG was used to obtain loss-delay sequences of UDP traffic. A little portion of the sequences was used as the training sequence to learn model parameters. Performance of trained model are tested on the remaining portions of the sequences.

In the following it is reported an example in which the inter-departure period is $T = 5 \cdot 10^{-3}\ s$ and the packet size is $N_b = 8 \cdot 10^3\ bits$, $(R = 1.6\ Mbps)$. The link was between the *D*ipartimento di Informatica e Sistemistica, Università di Napoli "Federico II", and the *D*ipartimento di Ingegneria dell'Informazione, Seconda Università di Napoli. A portion of the loss-delay sequence is shown in Fig. 5. Losses are indicated with negative values of y_n.

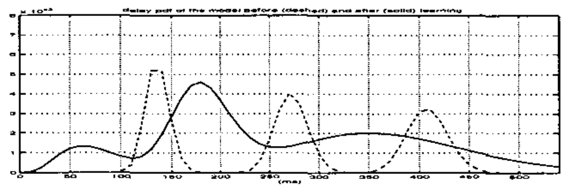
Training:

Fig. 6 shows typical trend of log-likelihood evolution during our EM learning procedure. We have used a training sequence of 1000 samples, which was used to train models with 2, 3, and 4 states. The algorithm convergence is reached after a few iterations. The HMM-based modelling appears to be a good strategy as it is able to capture both losses and delays characteristics quite well. Fig. 7 shows the delay pdf's before and after learning with 2−, 3−, and 4−state models, respectively in Figs. 7(b), 7(c), 7(d), in comparison to a delay histogram, in Fig. 7(a). The histogram clearly shows a multi-modal behavior. It is encouraging to see how well the model captures at increasing resolution the measured delay statistics as the number of hidden states is increased. Table 1 summarizes the results of the learning procedure in terms of the average loss probability (P_{loss}), showing how the trained models capture loss statistics too.

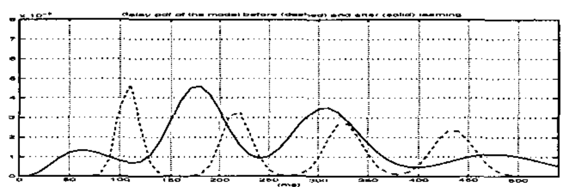
(a) Histogram of measured delays, i.e. of the positive value of training sequence y_n.



(b) Continuous term of pdf of a 2−state model before (dashed) and after (solid) learning procedure.



(c) Continuous term of pdf of a 3−state model before (dashed) and after (solid) learning procedure.



(d) Continuous term of pdf of a 4−state model before (dashed) and after (solid) learning procedure.

Fig. 7. Example of delays statistics learning for a 2−,3−, and 4−state model.

Model Generalization:

A trained model, to be useful, has to be tested on data which was not seen during training. Such generalization property has been verified for our model as it matches also future behavior of the channel. Fig. 8 shows the log-likelihood of the previous 2−, 3−, and 4−state trained models. Samples of a loss-delay sequence were grouped in blocks of 1000 consecutive samples. The first block constituted the training sequence while the other ones the test set, i.e. *test sequences*. The log-likelihood was evaluated for every test sequence. Circles and asterisks in Fig. 8 correspond to the log-likelihood for sequences evaluated respectively for starting and trained models. It can be noted how the trained models exhibit an almost constant log-likelihood (the same value as for the training sequence after training), showing how the channel can be considered to have stationary statistical characteristics for that time interval. Meanwhile the starting models exhibit lower and more

Table 1. Loss probability before and after the learning procedure compared to statistics of the training sequence.

	P_{loss}
training sequence	0.117
starting model	0.500
2 – state model	0.132
3 – state model	0.133
4 – state model	0.133

variable values for log-likelihood. Similar behaviors have been observed on different data sets obtained on different data links.

5. CONCLUSION

In this paper we have proposed a Bayesian Network whose objective is to model end-to-end packet channel behavior, jointly capturing losses and delays characteristics. The proposed model generalizes the HMM description of real channels introducing a joint stochastic modelling of losses and delays. Preliminary results are very encouraging, as the HMM is able to capture losses and delays characteristics of the network attributing automatically to various hidden states the dynamics of network congestion status. Future works will be focused on model improvements, predicting algorithms and content adaptation strategies.

6. REFERENCES

[1] E.N. Gilbert, *Capacity of a burst-noise channel.* Bell System Technical Journal, Vol. 39, pp. 1253-1265, Sept. 1960.

[2] E.O. Elliott, *Estimates of error-rate for codes on burst-noise channels.* Bell System Technical Journal, Vol. 42, pp. 1977-1997, Sept. 1963.

[3] L.A. Liporace, *Maximum Likelihood Estimation for Multivariate Observations of Markov Sources.* IEEE Transactions on Information Theory, Vol. IT-28(5), pp. 729-734, Sept. 1982.

[4] B.H. Juang, S.E. Levinson, and M.M. Sondhi, *Maximum Likelihood Estimation for Multivariate Mixture Observations of Markov Chains.* IEEE Transactions on Information Theory, Vol. IT-32(2), pp. 307-309, Mar. 1986.

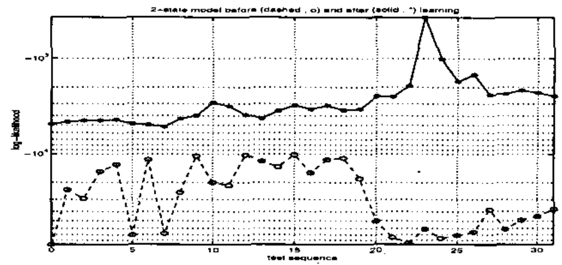
[5] L.R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* Proceedings of the IEEE, Vol. 77(2), pp. 257-285, Feb. 1989.

[6] J.C. Bolot, *Characterizing End-to-End Packet Delay and Loss in the Internet.* Journal of High-Speed Networks, Vol. 2(3), pp. 305-323, Dec. 1993.

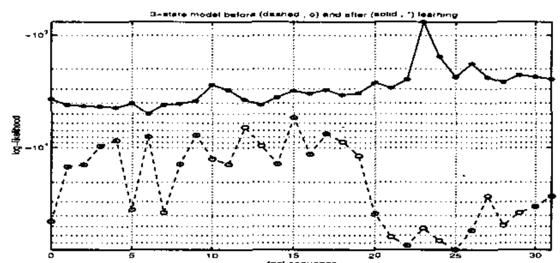
[7] M. Zorzi, R.R. Rao, and L.B. Milstein, *On the Accuracy of a First-Order Markov Model for Data Block Transmission on Fading Channels.* IEEE International Conference on Personal Communications, pp. 211-215, Nov. 1995.

[8] J.A. Bilmes, *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models.* Technical Report, ICSI-TR-97-021, University of Berkeley,CA, 1998.

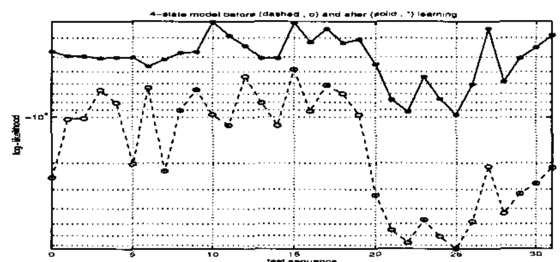
[9] V. Paxson, *End-toEnd Internet Packet Dynamics.* IEEE Transactions on Networking, Vol. 7(3), pp. 277-292, June 1999.



(a) Log-likelihood for a 2−state model before (dashed, ∘) and after (solid, ∗) learning procedure.



(b) Log-likelihood for a 3−state model before (dashed, ∘) and after (solid, ∗) learning procedure.



(c) Log-likelihood for a 4−state model before (dashed, ∘) and after (solid, ∗) learning procedure.

Fig. 8. Capacity of generalization of the model by use of log-likelihood.

[10] W. Jiang, and H. Schulzrinne, *Modeling of Packet Loss and Delay and Their Effect on Real-Time Mulrimedia Service Quality.* 10th International Workshop on Network and Operating System Support for Digital Audio and Video, June 2000.

[11] K. Salamatian, and S. Vaton, *Hidden Markov Modeling for Network Communication Channels.* ACM Sigmetrics/Performance, Vol. 29, pp. 92-101, 2001.

[12] J. Liu, I. Matta, and M. Crovella *End-to-End Inference of Loss Nature in a Hybrid Wired/Wireless Environment.* Technical Report, Boston University, , March 2002.

[13] S. Avallone, M. D'Arienzo, M. Esposito, A. Pescapé, S.P. Romano, and G. Ventre, *Mtools.* Networking Column on IEEE Network, Vol. 16(5), pp. 3, Sept./Oct. 2002.

[14] P. Salvo Rossi, G. Romano, F. Palmieri, and G. Iannello, *Bayesian Modelling for Packet Channels.* XIV Italian Workshop on Neural Nets, June 2003.