CONGESTION CONTROL FOR UDP TRAFFIC

Stefano Avallone (1), *Pierluigi Salvo Rossi* (1), *Vincenzo La Marca* (1)
Giulio Iannello (2), *Francesco Palmieri* (3), *Giorgio Ventre* (1)

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

ABSTRACT

Development of network-sensing and congestion-control mechanisms is one of the focus topic in the modern society of information as consequence of the huge growth in types of services, data links and connectivity. In this paper a monitoring system based on Hidden Markov Model is proposed and implemented such to interact with Distributed Internet Generator. Preliminary experiments performed in a real testbed shows excellent results in the ability to manage the source-traffic generation and adapt to the channel conditions.

1. INTRODUCTION

Modern communications requirements make network monitoring and adaptive strategies necessary tools to satisfy the huge demand of services. The growth in connectivity and in types of data links (wireless networks, satellites, cable modems, routers and all kinds of communication devices) will continue to make the global Internet a heterogeneous system very hard to analyze and manage. Too many details would be necessary to keep into account congestion, protocols, lossy links, noisy channels, etc., however a global equivalent end-to-end packet model, that could be easily estimated via low-speed two-way connections seems to be viable. The effectiveness of the model also depends on the level of stationarity of the network [7].

In this paper we present the implementation of a network-monitoring system, based on Hidden Markov Model (HMM), capable to estimate the current congestion level of the communication path between a sender and a receiver and to appropriately direct the sender to change the transmission rate, in order to minimize losses. The implemented system interacts with D-ITG traffic generator. Such traffic generator offers the possibility to send information on sent/received packets to external modules and to change "on-the-fly" the sender transmission rate. Such features are requested, respectively, to let the controller know about the network status and take appropriate actions.

Some experiments performed in a real testbed controlled by the HMM-based system are presented. They clarify the potential use of the model as a congestion-control mechanism of an adaptive communication scheme, as well as erroneous use derived from inappropriate parameters setting.

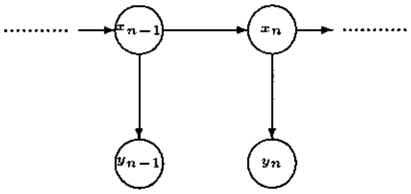


Fig. 1. Hidden Markov Model.

2. PACKET CHANNEL MODEL

An HMM that jointly describes losses and delays statistics has been proposed [9][10][13] to model end-to-end behavior for UDP traffic. The model is shown in Fig. 1, where $x_n \in \{s_0, s_2, \ldots, s_{N-1}\}$ denotes the hidden state variable related the the current congestion of the channel (s_i being the i^{th} state), and y_n denotes the observable variable that jointly represents losses and delays. Denote τ_n the delay of the n^{th} packet, then

$$y_n = \begin{cases} \tau_n & if \quad n^{th}\ packet\ is\ delivered \\ -1 & if \quad n^{th}\ packet\ is\ lost \end{cases} .$$

$\Lambda = \{\mathbf{A}, \mathbf{p}, \gamma, \vartheta\}$ is the set of parameters characterizing the model denoting the state transition matrix, the conditional loss probability vector and the conditional delay distribution[1] vectors respectively,

- $A_{ij} = Pr(x_{n+1} = s_j | x_n = s_i)$,
- $p_i = Pr(packet\ is\ loss | x_n = s_i)$,
- $\tau_n | x_n = s_i \sim Gamma(\gamma_i, \vartheta_i)$.

States can be classified on the basis of their conditional loss (p_i) and conditional mean delay ($d_i = \gamma_i \vartheta_i$). The average loss probability and the average delay of the model are $P_{loss} = \sum_{i=0}^{N-1} \pi_i p_i$ and $D_{mean} = \sum_{i=0}^{N-1} \pi_i d_i$ respectively, where π_i represents the steady-state probability.

Appropriate parameters for the model are found via the Baum-Welch algorithm [1][2][3], an iterative procedure searching for a local maximum of the likelihood of a trace used as training set. It

---

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)".

[1] Our choice of conditional probability density for delays is a Gamma distribution [4][5]

0-7803-8689-2/04/$20.00 ©2004 IEEE

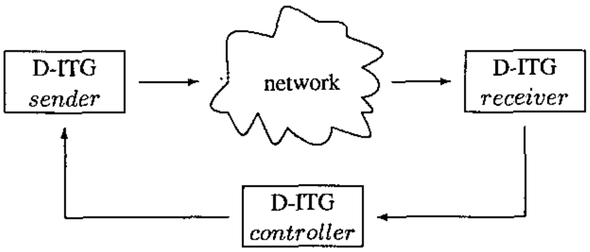


Fig. 2. Block diagram for the congestion management system.

is based on the following equations:

$$
\begin{aligned}
\hat{A}_{ij} &= \frac{\sum_{k=1}^{K-1} \alpha_k(i) A_{ij} b_j(y_{k+1}) \beta_{k+1}(j)}{\sum_{k=1}^{K-1} \alpha_k(i) \beta_k(i)} \\
\hat{p}_i &= \frac{\sum_{k=1}^{K} \rho_k(i) \beta_k(i)}{\sum_{k=1}^{K-1} \alpha_k(i) \beta_k(i)} \\
\hat{\gamma}_i \hat{\vartheta}_i &= \frac{\sum_{k=1}^{K} \rho_k(i) \beta_k(i) y_k}{\sum_{k=1}^{K-1} \rho_k(i) \beta_k(i)} \\
\hat{\gamma}_i \hat{\vartheta}_i^2 &= \frac{\sum_{k=1}^{K} \rho_k(i) \beta_k(i) (y_k - \mu_i)^2}{\sum_{k=1}^{K-1} \rho_k(i) \beta_k(i)}
\end{aligned}
$$

where

$$
\begin{aligned}
\alpha_k(j) &= \sum_{i=0}^{N-1} \alpha_{k-1}(i) A_{ij} b_j(y_k) \\
\beta_k(i) &= \sum_{j=0}^{N-1} A_{ij} b_j(y_{k+1}) \beta_{k+1}(j) \\
\rho_k(j) &= \sum_{i=0}^{N-1} \alpha_{k-1}(i) A_{ij} p_j \left. \frac{\partial b_j(t)}{\partial p_j} \right|_{t=y_k}
\end{aligned}
$$

3. DISTRIBUTED INTERNET TRAFFIC GENERATOR

The experiments described in Section 5 were conducted using D-ITG (Distributed Internet Traffic Generator) [8][11]. D-ITG generates both UDP and TCP traffic and measures both the one-way-delay and the round-trip-time. Information about throughput, packet loss, delay and jitter can be retrieved by processing (with an appropriate utility) the log files generated by the sender and the receiver. The architectural choices (such as the multi-threaded implementation) make D-ITG achieve high generation rates [12].

D-ITG implements various models to simulate sources of different application layer protocols (DNS, Telnet, VoIP, etc.). D-ITG allows for reproducibility of experiments: exactly the same pattern of traffic can be reproduced by setting the same seed value. It is also possible to set the Diffserv field and the Time-to-live field of the IP header.

One of the D-ITG features we exploited is the ability to remotely store log information. This means that the sender (and the receiver) can send log information to a log server instead of storing it locally. This feature has been thought for devices with limited storage capability, like PDAs, palmtops, etcetera.

Table 1. Average statistics of the 2- and 3-state models.

	2-state model	3-state model
P_{loss}	0.17	0.23
D_{mean}	265 ms	265 ms

Table 2. Conditional statistics of the 2- and 3-state models.

	2-state	3-state
p_0	0.13	0
p_1	0.24	0.08
p_2	–	0.82
d_0	251 ms	251 ms
d_1	274 ms	275 ms
d_2	–	264 ms

4. THE PROPOSAL

The HMM discussed in Section 2 was used to furnish D-ITG with a congestion-control mechanism. Fig. 2 shows the block diagram of the proposed architecture.

The D-ITG Sender sends traffic to the receiver, which does not locally store log information but sends it to the D-ITG controller. The latter component includes the functionalities of a log server. Thus, the D-ITG Controller receives information (timestamp, sequence number, size) about each received packet. This information is used to estimate the current state of the channel based on the Viterbi algorithm [3]. Not all packets are used in this calculation. The Viterbi algorithm is periodically invoked and uses a limited window of samples.

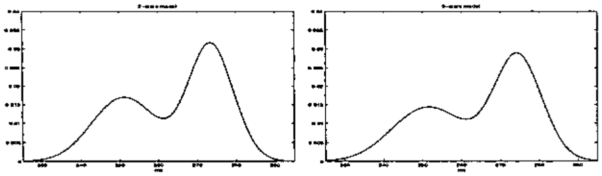
Based on the state determined by the Viterbi algorithm, D-ITG Controller may decide to increase or decrease the transmission rate of the sender. For this purpose, we added a new feature to D-ITG: the ability to change the transmission rate while generating traffic. This feature required some changes to the D-ITG sender, which must constantly listen for rate modification commands.

5. EXPERIMENTAL RESULTS

In this Section we present 3 experiments performed to investigate on the capability of the HMM-based software to manage network congestion. Each experiment lasted 300 s and the initial rate of the controlled traffic was set to 200 $pkts/s$. To avoid synchronization problems such as clock offset and skew experiments were performed using the round-trip-time mode, i.e. the D-ITG Receiver sends the packets back to the sender. It is the sender, therefore, to send log information to the D-ITG Controller. The controlled communication path thus consists of two parts, from the sender to the receiver and vice versa.

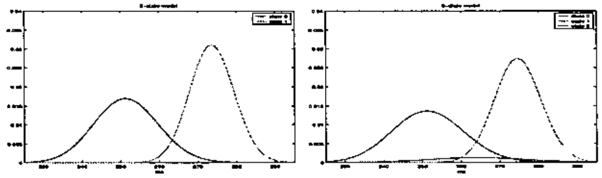
Congestion was caused by cross traffic that interferes with the controlled traffic. Cross traffic flows along the same communication path and is characterized by exponential inter-departure time and packet size.

Two models, with 2 and 3 states, has been learned off-line. Tabs. 1 and 2 show numerical values for average and conditional statistics respectively. Though the two models seem very similar, as confirmed also by Fig. 3 showing the continuous part of the model probability density (i.e. the distribution of delays), they have very different performance in congestion-network management.



(a) 2-state model. (b) 3-state model.

Fig. 3. Delay probability density.



(a) 2-state model. (b) 3-state model.

Fig. 4. Delay conditional probability density.

Looking carefully at the conditional statistics, Tab. 2 for loss and mean delay and Fig. 4 for delay distribution, it can be noted

- the 3-state model use two states (0 and 1) only to better match the bimodal behavior of delay statistics (they are loss-free) and one state (2) to account for losses,
- both the states of the 2-state model are used to account for loss and delay statistics, there is no separation in lossy and loss-free states.

The presence of lossy and loss-free states is very important for the performance of the congestion-control mechanism. Due to the memory and correlation existing between losses and delays, higher delays are usually observed in proximity of loss bursts [6], a model allowed to separate lossy and lossy-free behaviors can use this capability to prevent losses. Such a model introduce the "loss-prevention" state concept that turns to be useful for congestion control if an appropriate strategy depending on states of the model is chosen.

In the first experiment the Controller used the 2-state model, where s_0 and s_1 were considered "favorable" and "unfavorable" respectively. The strategy based on state estimation by the Controller for modifications of the Sender was:

- the Sender increases the rate of 10 $pkts/s$ if the Controller estimates s_0;
- the Sender decreases the rate of 10 $pkts/s$ if the Controller estimates s_1.

Fig. 5 shows the results in terms of rate at the Sender, estimated state at the Controller, packet loss and delay at the Receiver (remind that a loss is represented by -1), respectively. It can be noted how 2 states are not sufficient to manage appropriately the channel as also the "favorable" state is a lossy state. The transmission is characterized by high losses denoting the inappropriate rate of transmission of the Sender resulting in a waste of resource.

In the second experiment the Controller used the 3-state model, where s_2 was considered "unfavorable" being a lossy state, while the loss-free states s_0 and s_1 were considered "favorable" and "unfavorable" respectively, because of their different conditional mean delay, low the former and high the latter. The strategy based on state estimation by the Controller for modifications of the Sender was:

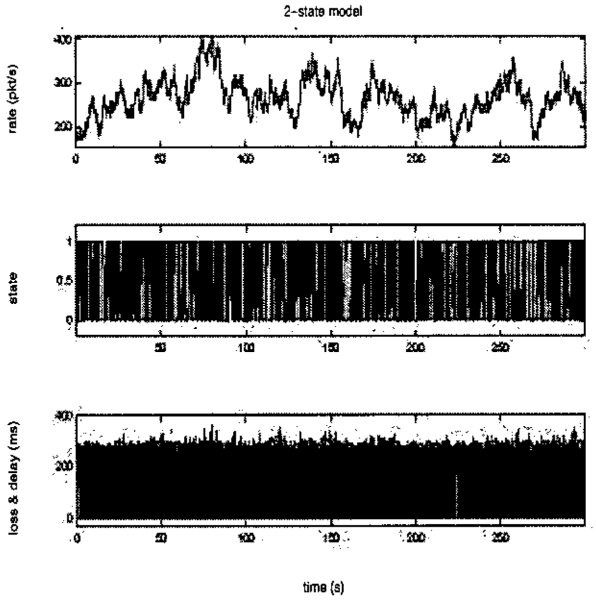


Fig. 5. Congestion control using a 2-state model.

- the Sender increases the rate of 10 $pkts/s$ if the Controller estimates s_0;
- the Sender decreases the rate of 10 $pkts/s$ if the Controller estimates s_1 or s_2.

Fig. 6 shows the results in terms of rate at the Sender, estimated state at the Controller, packet loss and delay at the Receiver. It can be noted how 3 states, appropriately used, allows an efficient use of the channel. This is due the fundamental role played by s_1, a free-loss state with high-delays, that can be viewed as a loss-prevention state. The action of the Controller is reducing the Sender rate until losses are not present. Then an oscillating behavior is induced as in absence of losses the Sender is forced to increase its rate, but the consequent increase of delay make the loss-prevention mechanism decrease the rate before losses occur.

To show the importance of the role of loss-prevention state, a third experiment is presented. The Controller used the 3-state model but with a different strategy, free-loss states s_0 and s_1 are considered "favorable" while lossy state s_2 is considered "unfavorable", i.e.

- the Sender increases the rate of 10 $pkts/s$ if the Controller estimates s_0 or s_1;
- the Sender decreases the rate of 10 $pkts/s$ if the Controller estimates s_2.

Fig. 7 shows the results in terms of rate at the Sender, estimated state at the Controller, packet loss and delay at the Receiver. In this case a 3-state model behaves like a 2-state model, resulting in high losses at the Receiver. This clarify the meaning of appropriate use of the 3 states, and the effectiveness of the loss-prevention state.

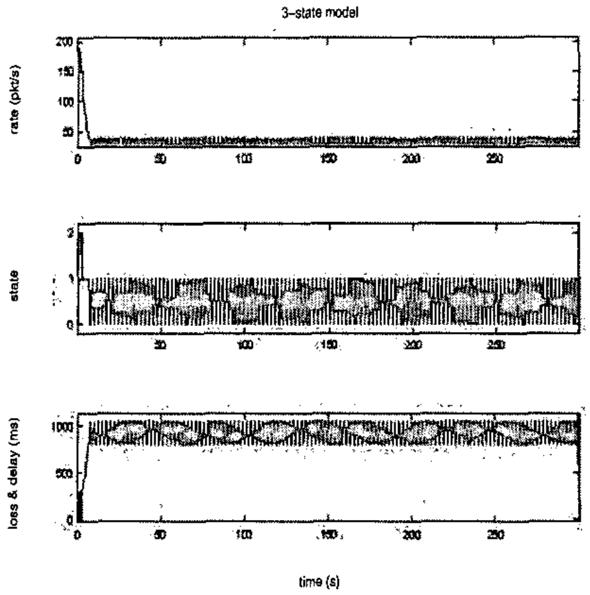


Fig. 6. Congestion control using a 3-state model appropriately.

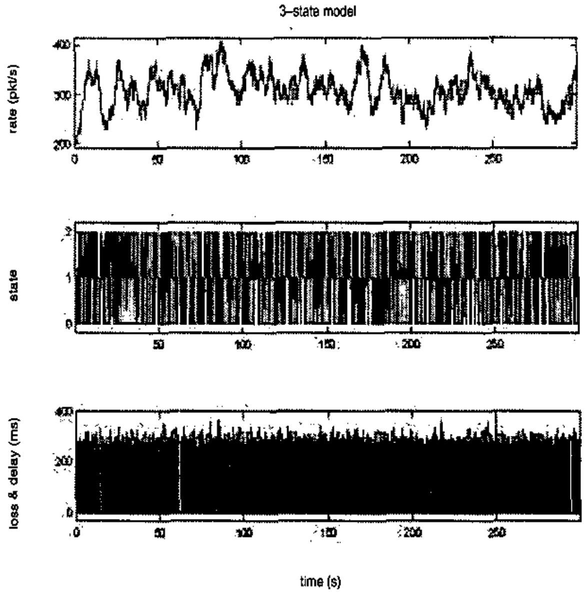


Fig. 7. Congestion control using a 3-state model not appropriately.

6. CONCLUSION

The proposal of an HMM-based network-sensing congestion-control mechanism is presented. It has been implemented as a software to be included in the D-ITG architecture. Preliminary results have been shown about its capability to manage the transmission rates depending on the network-path characteristics. The importance of properly interpreting the states of the model has also been underlined.

Future work will be focused on monitoring how long a learned model can be used. In the experiments presented in this paper, the cross traffic pattern generated during the on-line control phase is the same as that generated in the off-line learning phase. In a real scenario the characteristics of the path can change dramatically making necessary the re-estimation of the model parameters. Determining the conditions that should trigger a new learning phase is one of the most important goals. Furthermore the proposed scheme has to be tested and evaluated on different scenarios (links, interference traffic, etc.).

7. REFERENCES

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

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

[3] 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.

[4] 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.

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

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

[7] Y. Zhang, N. Duffield, V. Paxson, and S. Schenker, *On the Constancy of Internet Path Properties*. ACM SIGCOMM International Measurement Workshop, Nov. 2001.

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

[9] P. Salvo Rossi, G. Romano, F. Palmieri, and G. Iannello, *Bayesian Modelling for Packet Channels*. Italian Workshop on Neural Nets, Springer-Verlag, pp. 218-225, June 2003.

[10] P. Salvo Rossi, G. Romano, F. Palmieri, and G. Iannello, *A Hidden Markov Model for Internet Channels*. IEEE International Symposium on Signal Processing and Information Technology, Dec. 2003.

[11] S. Avallone, A. Pescapè, and G. Ventre, *Analysis and Experimentation of Internet Traffic Generator*. International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking, Feb. 2004.

[12] S. Avallone, D. Emma, A. Pescapè, and G. Ventre, *Performance evaluation of an open distribution platform for realistic traffic generation*. Special issue of Performance Evaluation: An International Journal (to appear in).

[13] P. Salvo Rossi, F. Palmieri, and G. Iannello, *HMM-based Monitoring of End-to-End Packet Channels*. IEEE International Conference on High Speed Network and Multimedia Communications, June 2004.

[14] D-ITG, Distributed Internet Traffic Generator. `http://www.grid.unina.it/software/ITG`