This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.

End-to-End Packet-Channel Bayesian Model applied to Heterogeneous Wireless Networks

Giulio Iannello
Univ. Campus Bio-Medico di Roma
Email: g.iannello@unicampus.it

Francesco Palmieri
Seconda Univ. di Napoli
Email: francesco.palmieri@unina2.it

Antonio Pescapè and Pierluigi Salvo Rossi
Univ. di Napoli "Federico II"
Email: {pescape,salvoros}@unina.it

***Abstract*— This paper proposes a source-traffic based model to estimate jointly packet losses and delays statistical behavior of a network path. The approach relies on a Hidden Markov Model built on real-traffic information. The effectiveness of the model is evaluated over different real heterogeneous network scenarios. Our experimental results show that the model captures average (long-term) and conditional (short-term) statistics that in most cases are typical of the single scenario. Preliminary results about prediction on a sample path as well as investigation on the use of the same model across different scenarios are given.**

I. Introduction

Stationary phenomena in the global Internet make network monitoring and adaptive strategies quite appealing for better understanding and management [1]. Testing analytical models on different scenarios is crucial for effective adaptive-techniques design.

Our analysis focuses on equivalent models for packets loss/delay end-to-end behavior. Loss phenomena often show *bursty behavior*, and packet delays present memory [2][3][4]. Furthermore, in real networks, losses and delays are strongly correlated as it has been observed that in proximity of a loss, larger delays tend to occur [4].

The works of Gilbert and Elliott [5][6] on bit-transmission burst-error channels showed how a 2-state Hidden Markov Model (HMM) was effective in characterizing some real channels. Analogously end-to-end packet channels show burst-loss behavior. Jiang and Schulzrinne [4] investigated packet-channels behavior finding that Markov models are not appopriate for inter-loss behavior and that delays manifest temporal dependency. Salamatian and Vaton [7] found that an HMM trained with experimental data captures loss behavior, and HMMs with up to 4 states fit well the data. Liu, Matta and Crovella [8] used an HMM-based loss-delay modeling of TCP traffic in order to infer loss nature in hybrid wired/wireless environments, showing how it can be used to control TCP congestion avoidance mechanism. Similar works have been done by Turin, Sondhi, Zorzi, Rao, Milstein [9][10] on error sources for digital channels and wireless fading links. Wei, Wang, Towsley and Kurose [11] proposed Markov-based models to infer from End-to-End measurements the knowledge about the presence in the path of a dominant congested link. Tao and Guérin [12] faced the problem of on-line learning and prediction of loss behavior via HMMs built on probe traffic and inferring congestion properties in terms of packet loss. A continuous-time HMM is derived and used for a general traffic source, but the assumption that statistics learned by the probe traffic hold on for the real network behavior is needed.

These works suggest that a Bayesian state-conditioned model may be effective in capturing loss/delay dynamic behavior on packet channels. What emerges is that different kinds of Markov-based models allow appropriate modeling of variegate network environment while keeping mathematical tractability. Our approach is a joint description of loss and delay based on HMM in order to capture their memory and correlation, obtained by appropriate introduction of a hidden variable related to the current channel congestion.

The proposed model showed to be effective for UDP traffic over wired connections [13]. Here we test the applicability over several heterogeneous network scenarios. State knowledge and prediction may enable a powerful characterization of future channel behavior, which could be used to implement content-adaptive strategies for coding (e.g., Multiple Description Coding) and scheduling (e.g. traffic shaping). Some preliminary considerations are given for the TCP case.

To highlight the significance of the proposed approach we underline that: (i) it allows loss/delay joint description; (ii) it allows on-line learning; (iii) it allows network monitoring; (iv) it allows prediction; (v) it is derived by source traffic (i.e. no probe traffic overhead is needed); (vi) it explicitly takes into account the influence of source traffic over the network in order to avoid assumption on network behavior; (vii) it allows to distinguish loss/delay nature; (viii) it has been tested on a wide range of different heterogeneous wireless scenarios and different types of traffic, deriving analogies and differences on the equivalent end-to-end models.

II. The Analytical Model

Fig. 1 shows the reference end-to-end packet channel, including the network and the underlying protocol layers between a source-destination pair. Numbered constant-size packets are transmitted, $t_n^{'}$, $t_n^{''}$, Δ_n and τ_n denote the departure time, the arrival time, the inter-departure time (IDT) and the accumulated delay of the n^{th} packet, respectively, i.e. $\Delta_n = t_n^{'} - t_{n-1}^{'}$ and $\tau_n = t_n^{''} - t_n^{'}$. The network randomly drops and delays packets according to the current congestion. Loss/delay memory and correlation are taken into account introducing

[0]This work has been partially supported by the MIUR in the framework of the WEB-MINDS FIRB Project.

Fig. 1. End-to-end packet channel

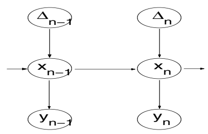


Fig. 2. Structure of an IO-HMM

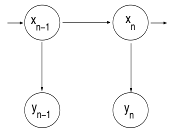


Fig. 3. Structure of an HMM

a hidden state variable (that may be inferred by loss/delay observation) related to the network-path congestion. Furthermore, it may be reasonable that source-traffic characteristics influence the channel behavior. Traffic flows with different bit-rates experience different loss/delay patterns on the same end-to-end channel. Source-traffic characteristics has to be taken into account in the objective of channel modeling.

Let us denote y_n a variable describing the outcome experience of the n^{th} packet, ($y_n > 0$ means $\tau_n = y_n$, while $y_n = -1$ means the n^{th} has been lost), and $x_n \in \{s_1, s_2, \ldots, s_N\}$ a variable distinguishing the state of the link at time $t_n^{'}$ among N possible ones. We assume the state variable follows a Markov chain structure, more specifically loss/delay statistics are described via an IO-HMM [14], shown in Fig. 2, where Δ_n, x_n and y_n are the input, the hidden-state and the output variables, respectively. The set of parameters characterizing the model is $\lambda = \{\mathbf{A}(.), \mathbf{p}, \gamma, \vartheta\}$, where

- $\mathbf{A}(u) = [A_{i,j}(u)]_{i,j=1}^{N}$ is the state transition matrix, i.e. $A_{i,j}(u) = \Pr(x_{n+1} = s_j | x_n = s_i, \Delta_n = u)$,
- $\mathbf{p} = [p_1, \ldots, p_N]^T$ is the conditional loss probability vector, i.e. $p_i = \Pr(n^{\text{th}}$ packet is lost $|x_n = s_i)$,
- $\gamma = [\gamma_1, \ldots, \gamma_N]^T$ and $\vartheta = [\vartheta_1, \ldots, \vartheta_N]^T$ are the conditional delay vectors, in the state s_i delays are Gamma distributed with parameters $\{\gamma_i, \vartheta_i\}$,
- y_n is a hybrid random variable whose conditional probability density function (pdf) is
$B_i(y) = p_i \delta(t+1) + (1-p_i) \frac{(\frac{y}{\vartheta_i})^{\gamma_i - 1} e^{-\left(\frac{y}{\vartheta_i}\right)}}{\vartheta_i \Gamma(\gamma_i)}$,
where $\delta(.)$ and $\Gamma(.)$ denotes the Dirac impulse and the Gamma function, respectively.

The choice of delay distribution has been based on [3][15] even though such an assumption for wireless networks it is not obvious.

The average loss probability and delay of the model are

$$P_{\text{loss}} = \sum_{i=1}^{N} \pi_i p_i \quad , \quad D_{\text{mean}} = \sum_{i=1}^{N} \pi_i d_i \ , \tag{1}$$

respectively, where $\pi = [\pi_1, \ldots, \pi_N]^T$ is the steady-state probability distribution, i.e

$$\pi_i = \int_0^{+\infty} \pi_i(u) f_\Delta(u) du \ , \ \mathbf{A}^T(u)\pi(u) = \pi(u) \ ,$$

being $f_\Delta(.)$ the IDT pdf, and

$$d_i = \gamma_i \vartheta_i \ , \tag{2}$$

the conditional mean delay.

The model throughput $\eta(\tau_{\max})$, the fraction of delivered packets within a maximum allowed delay $\tau_{\max}$, is

$$\eta(\tau_{max}) = \sum_{i=1}^{N} \pi_i (1-p_i) \Gamma(\frac{\tau_{\max}}{\vartheta_i}, \gamma_i) \ .$$

The model auto-correlation $R(m) = E\{y_n, y_{n+m}\}$ is[1]

$$R(m) = \begin{cases} \pi^T \ \mathbf{E}_{II} \ \mathbf{1} & m = 0 \\ \pi^T \ \mathbf{E} \ \mathbf{Q}^{m-1} \ \mathbf{E} \ \mathbf{1} & m > 0 \\ R(-m) & m < 0 \end{cases} .$$

The model simply reduces to an HMM [16][13], shown in Fig. 3, if packets are transmitted with constant IDT, $t_n^{'} = nT$ and $\Delta_n = T$.

A. Learning Procedure

Appropriate model parameters has to be found to model a real channel. The EM algorithm[2] [16] allows to find the most likely set of parameters matching the statistics of a set of output (and input in case of IO-HMM's) sequences, namely the *training set*. It is an iterative procedure searching for a local maximum, with respect to the parameters of the model, of the likelihood of the training set $\mathcal{L}(\lambda) = \Pr(\mathbf{y}|\mathbf{\Delta}, \lambda)$ [3]. It is based on recursive computation of appropriate *Forward* and *Backward* variables. In the following, the set of parameters that represent the initialization of the procedure will be referred to as the *starting model*. The *Forward* and *Backward* variables for an IO-HMM are defined, respectively, as

$$\begin{cases} \alpha_n(i) = \Pr(y_1, \ldots, y_n, x_n = s_i | \Delta_1, \ldots, \Delta_n, \lambda) \\ \beta_n(j) = \Pr(y_{n+1}, \ldots, y_L | x_n = s_j, \Delta_{n+1}, \ldots, \Delta_L, \lambda) \end{cases} .$$

They can be recursively computed by use of

$$\begin{cases} \alpha_{n+1}(j) = \sum_{i=1}^{N} \alpha_n(i) A_{i,j}(\Delta_{n+1}) B_j(y_{n+1}) \\ \beta_{n-1}(i) = \sum_{j=1}^{N} A_{i,j}(\Delta_n) \beta_n(j) B_j(y_n) \end{cases} .$$

The re-estimation formulas of the learning algorithm are based on the following quantities,

$$\begin{aligned} \xi_n(i,j) &= \Pr(x_n = s_i, x_{n+1} = s_j | \mathbf{y}, \mathbf{\Delta}, \lambda) \\ &= \frac{\alpha_n(i) A_{i,j}(\Delta_{n+1}) \beta_{n+1}(j) B_j(y_{n+1})}{\mathcal{L}(\lambda)} \ , \end{aligned}$$

$$\gamma_n(i) = \Pr(x_n = s_i | \mathbf{y}, \mathbf{\Delta}, \lambda) = \frac{\alpha_n(i)\beta_n(i)}{\mathcal{L}(\lambda)} \ ,$$

where $\mathcal{L}(\lambda) = \Pr(\mathbf{y}|\mathbf{\Delta}, \lambda) = \sum_{i=1}^{N} \alpha_n(i) \beta_n(i)$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_n(i) A_{i,j}(\Delta_{n+1}) \beta_{n+1}(j) B_j(y_{n+1}) \ . \tag{3}$$

[1] $Q_{i,j} = \int_0^{+\infty} A_{i,j}(u) f_\Delta(u) du$; $E_{i,j} = Q_{i,i}\left[(1-p_i)\gamma_i\vartheta_i - p_i\right]\delta_{i,j}$; $E_{II\ i,j} = Q_{i,i}\left[(1-p_i)(1+\gamma_i)\gamma_i\vartheta_i^2 + p_i\right]\delta_{i,j}$; and $\mathbf{1}$ is a vector whose elements are 1.

[2] It reduces to the *Baum-Welch* algorithm for HMM's structures.

[3] $\left\{\mathbf{\Delta} = [\Delta_1, \ldots, \Delta_L]^T, \mathbf{y} = [y_1, \ldots, y_L]^T\right\}$ is the training set.

Denoting $\rho_n(j) = \sum_{i=1}^{N} \alpha_{n-1}(i) A_{i,j}(\Delta_n) p_j \left.\frac{\partial b_j(t)}{\partial p_j}\right|_{t=y_n}$, the set of parameters is updated each iteration by use of the following formulas,

$$\hat{A}_{i,j}(u) = \frac{\sum_{\substack{n=1\\ \Delta_{n+1}=u}}^{L-1} \xi_n(i,j)}{\sum_{\substack{n=1\\ \Delta_{n+1}=u}}^{L-1} \gamma_n(i)},$$

$$\hat{p}_i = \frac{\sum_{n=1}^{L} \rho_n(i)\beta_n(i)}{\sum_{n=1}^{L} \gamma_n(i)},$$

$$\hat{\gamma}_i\hat{\vartheta}_i = \frac{\sum_{n=1}^{L} \rho_n(i)\beta_n(i) y_n}{\sum_{n=1}^{L} \rho_n(i)\beta_n(i)},$$

$$\hat{\gamma}_i\hat{\vartheta}_i^2 = \frac{\sum_{n=1}^{L} \rho_n(i)\beta_n(i)(y_n - \gamma_i\vartheta_i)^2}{\sum_{n=1}^{L} \rho_n(i)\beta_n(i)}.$$

III. Experimental Results

We conducted empirical experiments, analyzing the ability of the model to capture system invariances with respect to packet loss/delay in various heterogeneous scenarios. We considered heterogeneous networks in terms of *access network technologies* (ANTs), *end-users devices*, *Operating Systems* (OS) and *end-users application*. Due to space limitations, in this paper we have focused on 802.11b in Ad-Hoc configuration, GPRS, UMTS, and Ethernet. The performance evaluation study has been performed using D-ITG (*Distributed Internet Traffic Generator* [17]. We have proceeded obtaining empirical data from the different scenarios[4] to be processed by the model described in Section II. Output sequences[5] are derived from packet numbers, transmission and delivery times, available by use of D-ITG. The results showed in this paper are related to situations with *high* traffic load relying on the channel, exhibiting the network a strong burstiness that makes i.i.d. modeling rather inadequate.

More precisely, in this paper we show results when we applied the model to the following scenarios:

(A) Ad-Hoc networks where we have Laptops, Linux OS, UDP, and a WLAN 802.11b as ANT;

(B) Integration of LANs and GPRS networks (Ethernet/WLAN 802.11b and GPRS) where we have Laptops, both Linux and Windows XP OS, UDP, and Ethernet, WLAN 802.11b, and GPRS as ANT;

(C) Integration of LANs and UMTS networks (Ethernet/WLAN 802.11b and UMTS) where we have Laptops, both Linux and Windows XP OS, UDP, and Ethernet, WLAN 802.11b, and UMTS as ANT.

The reason of our choice relies on the following considerations. Currently in real heterogeneous scenarios a demand to integrate WLANs with third-generation (3G) cellular networks, such as GSM/GPRS, UMTS has yielded. In these environments, among the many factors that determine the feasibility of a given network scenario for the given set of application requirements, there is the possibility to have a realistic and innovative network model characterizing the performance.

[4]It is worth noting that we ignored data related to the initial and final transients, otherwise they would have affected uselessly the learned statistics.

[5]Sequences carrying information about losses and delays of source traffic.

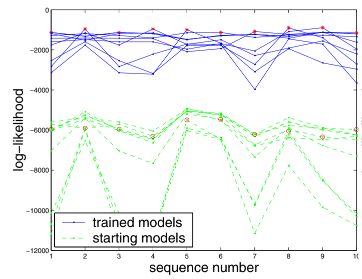


Fig. 4. Ad-Hoc configuration.

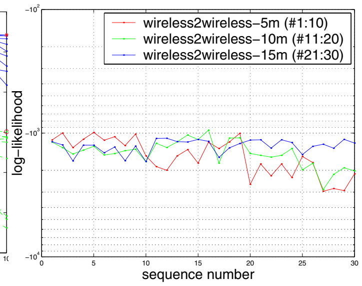


Fig. 5. Models comparison.

TABLE I

Average and conditional statistics of 2-state models.

	P_{loss}	D_{mean}	p_1	p_2	d_1	d_2
1)	0.91	0.136s	0.94	0.89	0.146s	0.125s
2)	0.91	0.133s	0.94	0.88	0.144s	0.122s
3)	0.91	0.130s	0.94	0.89	0.141s	0.121s

A. Ad-Hoc networks

The first step of our analysis has been to evaluate how significant is a measurement relative to a given configuration. In order to address this point, a group of 10 different measurements and then 10 output sequences has been considered for each configuration. A model has been trained on each sequence and then tested on the remaining 9 sequences of the same group. The test consists in computing the likelihood (Eq. (3)) of the test sequences given the model.

Fig. 4 shows the results for a configuration with $10\text{ m} \leq d \leq 15\text{ m}$, UDP traffic with IDT of 0.1 ms and packet size (PS) of 128 bytes. The blue solid lines represent the log-likelihood of the 10 trained models on the 10 output sequences (9 test sequences and 1 training one with a red asterisk), the green dashed lines refer to the starting models[6].

The almost stable behavior of the trained models assures that a single measurement and the derived trained model well represents the particular configuration. Similar results have been obtained over different configurations. This allows to consider *one model per configuration* in order to analyze analogies and differences among different scenarios. Fig. 5 considers 3 models trained on 3 different configurations and tested on the 30 sequences from the groups themselves: (i) $0\text{ m} \leq d \leq 5\text{ m}$, UDP traffic with IDT of 0.1 ms and PS of 128 bytes; (ii) $5\text{ m} \leq d \leq 10\text{ m}$, UDP traffic with IDT of 0.1 ms and PS of 128 bytes; $10\text{ m} \leq d \leq 15\text{ m}$, UDP traffic with IDT of 0.1 ms and PS of 128 bytes.

The red, green, and blue lines refer to the 3 models, respectively. Sequence numbers from 1 to 10, 11 to 20, and 21 to 30, refer to sequences from the 3 groups, respectively. It can be noted how the 3 configurations have a similar behavior.

[6]It is worth noting that the starting models have been chosen in order to present the correct average statistics.

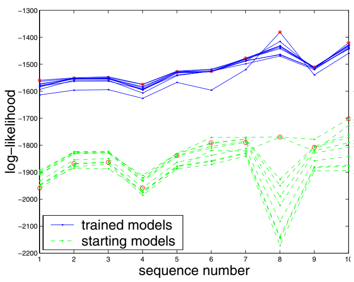


Fig. 6. LANs-GPRS.

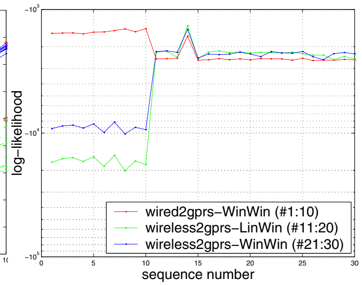


Fig. 7. Models comparison.

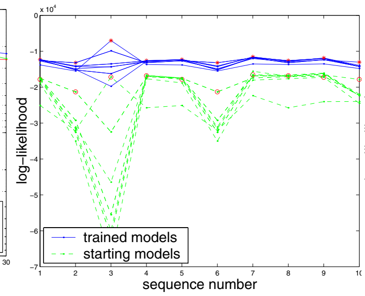


Fig. 8. UMTS-GPRS.

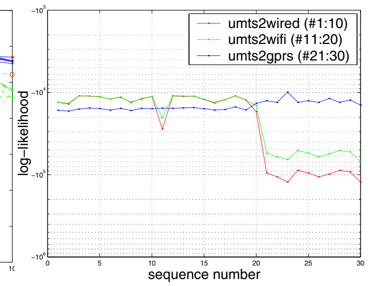


Fig. 9. Models comparison.

TABLE II

AVERAGE AND CONDITIONAL STATISTICS OF 2-STATE MODELS.

	P_{loss}	D_{mean}	p_1	p_2	d_1	d_2
1)	0.34	4.29s	0.05	0.60	5.91s	2.51s
2)	0.90	1.76s	0.92	0.88	2.00s	1.52s
3)	0.89	1.78s	0.90	0.88	1.92s	1.45s

TABLE III

AVERAGE AND CONDITIONAL STATISTICS OF 2-STATE MODELS.

	P_{loss}	D_{mean}	p_1	p_2	d_1	d_2
1)	0.13	1.52s	0.43	0.001	1.90s	1.33s
2)	0.15	1.79s	0.48	0.001	2.45s	1.45s
3)	0.25	5.17s	0.09	0.37	7.99s	3.68s

This is confirmed if we look at the statistics of the configurations. Tab. I shows average and conditional statistics (Eqs. (1) and (2)) for the 3 configurations, respectively (we considered 2-state models). It can be noted how the similarity involves not only average statistics, but even conditional, i.e. they have similar local behaviors. More precisely, the rows are related to the 3 configurations previously enumerated.

B. Integration of LANs and GPRS networks

The same considerations on the significance of a measurement about a given configuration hold for this new scenario. Fig. 6 shows the results in the case of integration of LANs and GPRS networks, for UDP traffic with IDT of 10 ms and PS of 512 bytes. Fig. 7 considers 3 models trained on 3 different configurations and tested on the 30 sequences from the groups themselves: (i) *Wired2GPRS scenario*: Laptop, Windows XP OS, UDP traffic with IDT of 10 ms and PS of 512 bytes, Ethernet ANT at sender side whereas Laptop, Windows XP OS, GPRS ANT at receiver side; (ii) *WiFiL2GPRS scenario*: Laptop, Linux OS, UDP traffic with IDT of 10 ms and PS of 512 bytes, WLAN 802.11b ANT at sender side whereas Laptop, Windows XP OS, GPRS ANT at receiver side; (iii) *WiFiW2GPRS scenario*: Laptop, Windows XP OS, UDP traffic with IDT of 10 ms and PS of 512 bytes, WLAN 802.11b ANT at sender side whereas Laptop, Windows XP OS, GPRS ANT at receiver side. It can be noted how the second and the third configurations have a similar behavior, while the first has a different one. This is due to the presence of the wired Ethernet channel.

Again this is confirmed via Tab. II showing average and conditional statistics (Eqs. (1) and (2)) for the 3 configurations (we considered 2-state models). It can be noted how the similarity between the second and the third configurations and the differences with the first one, involves not only average statistics, but even conditional, that is they have very different temporal local behaviors. Measurements from the second and third configurations exhibit a high-delay state (state 1) with higher losses than the low-delay state (state 2), while measurements from the first configuration exhibits an opposite behavior, i.e. high-delay state present lower losses than low-delay state.

C. Integration of LANs and UMTS networks

Fig. 8 shows the results in the case of integration of LANs and UMTS networks for UDP traffic with IDT of 10 ms and PS of 64 bytes. Fig. 9 considers 3 models trained on 3 different configurations and tested on the 30 sequences from the groups themselves: (i) *UMTS2Ethernet scenario*: Laptop, Windows XP OS, UDP traffic with IDT of 10 ms and PS of 64 bytes, UMTS ANT at sender side whereas Laptop, Windows XP OS, Ethernet ANT at receiver side; (ii) *UMTS2WiFi scenario*: Laptop, Linux OS, UDP traffic with IDT of 10 ms and PS of 64 bytes, UMTS ANT at sender side whereas Laptop, Windows XP OS, WLAN 802.11b ANT at receiver side; (iii) *UMTS2GPRS scenario*: Laptop, Windows XP OS, UDP traffic with IDT of 10 ms and PS of 64 bytes, UMTS ANT at sender side whereas Laptop, Windows XP OS, GPRS ANT at receiver side. It can be noted how the first and the second configurations have a similar behavior, while the third has a different one. Indeed, in the first two situations the perceived performance are influenced by the presence of the UMTS network.

Again this is confirmed via Tab. III showing average and conditional statistics (Eqs. (1) and (2)) for the 3 configurations (we considered 2-state models), showing differences and similarities both on average and conditional statistics. Measurements from the first and second configurations exhibit a high-delay state (state 1) with higher losses than the low-delay state (state 2), while measurements from the third configuration exhibits an opposite behavior, i.e. high-delay state present lower losses than low-delay state.

D. Discussion

Comprehension about when it is possible to use a trained model is crucial to determine its generalization capability.

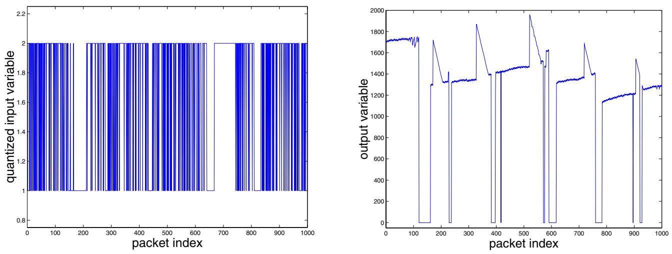


Fig. 10. Pareto traffic pattern.

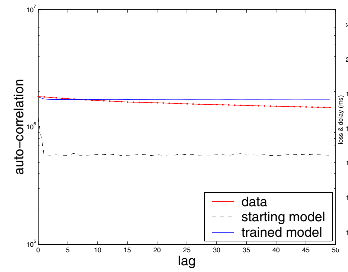


Fig. 11. Pareto traffic correlation.

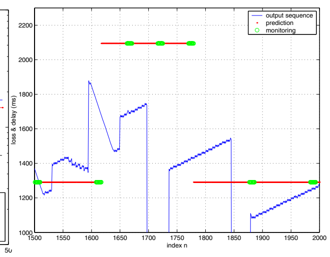


Fig. 12. Prediction.

The heterogeneous characteristics of real networks may be assumed as sequences of almost stationary periods, sometimes needing parameters re-estimation. An adaptive protocol limiting the resource spent for its training is desirable. The previous analysis reveals that: (i) each configuration presents stable characteristics, both long- and short-term (average and conditional statistics); (ii) components of different scenarios like terminal distance, OS, etc. do not seem to influence the equivalent channel but scaling numerical values; (iii) things are different for access-point and/or end-point typology, etc. that modify channel behavior more significantly, altering loss-delay cross-correlation. Stable characteristics of various configurations testify that the model is well posed and can work also in wireless environments. The analysis shows that the model works by distinguishing different modes (short-term behavior) on the basis of loss/delay characteristics in the equivalent end-to-end channel, and it does not depend on the nature of the links implementing the channel (wired, wireless, etc.). Whenever the equivalent behavior is characterized by a mixture of simpler modes, the model is able to find them out, compatibly with the number of hidden states it has been furnished. No matter if the bursty behavior is due to buffer overflows, deep fading or whatever else, the models only cares about how they affect loss/delay statistical behavior[7]. However different modes are typically related to different phenomena, i.e. the model allows to infer loss/delay nature (wireless, congestion, etc.) via state estimate.

E. Analyzing Pareto Traffic Patterns

The model was also used on UDP traffic with Pareto and Exponential distributed IDT. To simplify the training procedure we considered a quantized version of $\mathbf{\Delta} = [\Delta_1, \ldots, \Delta_L]^T$ as input sequence[8]. In this case we took into account UMTS ANT, Windows XP OS, UDP traffic with PS of 64 bytes Laptop at sender side whereas WLAN 802.11b ANT, Windows XP OS, Laptop at receiver side. Fig. 10 shows an example of input and output sequences for Pareto UDP traffic with shape and scale parameters $\alpha_\Delta = 15$ and $\beta_\Delta = 9$ ms. The input sequence has been quantized with two levels, 1 for $\Delta_n < \frac{\alpha_\Delta \beta_\Delta}{\alpha_\Delta - 1}$ and 2 for $\Delta_n \geq \frac{\alpha_\Delta \beta_\Delta}{\alpha_\Delta - 1}$. It can be seen how the channel behavior is strongly influenced by the input sequence. Long sequences of 1's in the input sequence (high bit-rate) induce a very lossy behavior to the channel. This dependence strengthens the proposal to account for variable source traffic and reveals very appropriate when the source traffic cannot be assumed a neglecting portion of the total one.

Fig. 11 shows the autocorrelation of the starting and trained models compared to the auto-correlation of a Pareto UDP trace. This example shows how the IO-HMM is the appropriate generalization of the HMM previously considered. Similar results to previous scenario have been obtained for various non-periodic UDP traffic typologies.

F. Predicting the network behavior

The trained models have been used for prediction purposes on a same path. Testing prediction capability is a necessary step for the evaluation of the model effectiveness if thought as a component of a more complex network-sensing and adaptive communication protocol. The network-sensing, the prediction and the adaptive steps are to be based on current-state estimate. State estimated by use of the Viterbi algorithm [16] has shown to be well representative of the congestion level of the path [13] showing sufficient local stability. Furthermore, it is worth noting that states can be labeled as "favorable" and "unfavorable" basing on their conditional statistics (p_i and d_i) allowing definition of appropriate adaptive strategies. Preliminary tests have been performed showing good results. We considered a single scenario, the corresponding trained model, and different output sequences. The output sequences have been iteratively processed by the model such that:

sensing: W samples of the output sequence are used to estimate the current network state by use of the Viterbi algorithm [16];

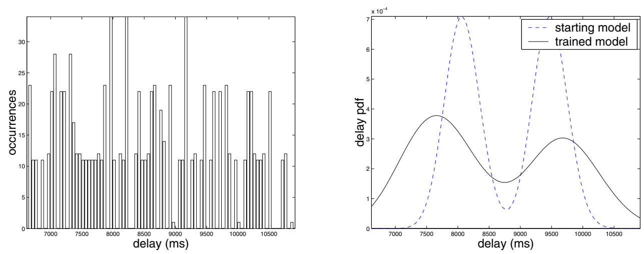
prediction: the network is assumed to behave as being in the estimated state (i.e. with average statistics given by the corresponding conditional ones) for a time proportional to the average duration of the state[9].

Fig. 12 shows a portion of a trace related to the first configuration presented in Section III-C. Conditional mean delays are plotted to denote estimated and predicted states. The blue-line refers to the output sequence (packet loss/delay),

[7]We do not address the problem of studying the sensibility of the model depending on the PS of the source traffic. Most likely different packet-size traffics experience totally different paths, considering a unique equivalent end-to-end model is not appropriate in our opinion.

[8]This allow to avoid dealing with softmax functions in the training procedure [14] being $\mathbf{A}$ a 3-dimensional matrix.

[9]The average duration in state x_i is given by $\frac{1}{1-A_{i,i}}$.



(a) Histogram of the delays. (b) Delay statistics of the models.

Fig. 13. Training the model for TCP traffic.

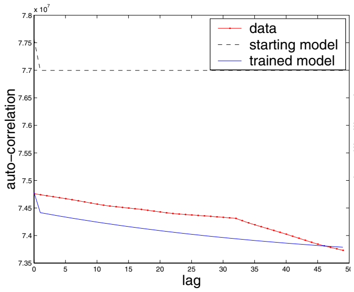


Fig. 14. TCP traffic correlation.

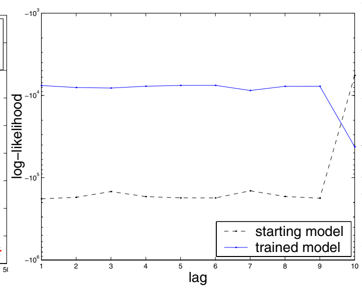


Fig. 15. Test on TCP patterns.

the green circles refer to data used for Viterbi state estimate while the red points represent the predicted states. However many details have not been analyzed yet, such as dependence on the length of the window for the state estimate, influence of the type of prediction, definition of the adaptive strategies, etc. Similar results have been obtained for other scenarios.

G. The proposed model in the TCP scenario

The proposed model has been tested also on TCP traces. We have taken into account WLAN 802.11b ANT, Linux OS, TCP traffic with IDT of 10 ms and PS of 128 bytes, Laptop at sender side whereas GPRS ANT, Windows XP OS, Laptop at receiver side. Obviously in this case we have taken into account only network delays ($\mathbf{p}$ is the null vector). The training procedure captures delay statistics also for the case of TCP, Fig. 13 shows the histogram of the delays of the output sequence used as training set and the pdf of delays of the starting and trained models. Though it could seem that the models before and after learning procedure are very similar, the difference can be appreciated looking at the delay auto-correlation. Fig. 14 shows via the auto-correlation of the starting and trained models and of the output sequence how the the data behavior is well captured. The effectiveness of the trained model has been tested on different output sequences from the same group of measurement. In Fig. 15 the likelihood of the starting and of the trained models is shown for a group of 10 sequences, where every point represent a TCP output sequence. It turns out that also for TCP data a trained model holds his generalization capability. The recent interest for Bayesian technique in TCP scenario relies on the ability to better infer channel state from loss/delay observation [18].

IV. Conclusion

In this work we presented an HMM-based packet-channel analytical model applied to heterogeneous wireless networks and its validation. The model showed to capture network behavior quite well and to predict packet loss/delay statistics. We tested the model over a wide range of network scenarios, showing results related to Ad-Hoc networks, integration of GPRS and WLAN 802.11b, and integration of UMTS and WLAN 802.11b. The experiments showed how components of different scenarios play different roles in the equivalent end-to-end channel model, some inducing a simple scaling in the model parameters, others dramatically changing the correlation structure of loss/delay phenomena as well as the short-term behavior of the network. The model has been also tested in its capability to take into account explicitly the influence of the source traffic on the network. Preliminary tests have been performed on TCP traces showing encouraging results. Finally tests on prediction capability have been performed to ensure about its effectiveness, though they are only a first-step verification (being performed on off-line measurements).

References

[1] Y. Zhang, N. Duffield, V. Paxson, S. Schenker, “On the Costancy of Internet Path Properties”, *ACM SIGCOMM IMW*, Nov. 2001.
[2] 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.
[3] V. Paxson, “End-to-End Internet Packet Dynamics”, *IEEE Trans. on Networking*, Vol. 7(3), pp. 277-292, June 1999.
[4] W. Jiang, H. Schulzrinne, “Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality”, *NOSSDAV*, June 2000.
[5] E.N. Gilbert, “Capacity of a burst-noise channel”, *Bell System Technical Journal*, Vol. 39, pp. 1253-1265, Sept. 1960.
[6] E.O. Elliott, “Estimates of error-rate for codes on burst-noise channels”, *Bell System Technical Journal*, Vol. 42, pp. 1977-1997, Sept. 1963.
[7] K. Salamatian, S. Vaton, “Hidden Markov Modeling for Network Communication Channels”, *ACM Sigmetrics/Performance*, Vol. 29, pp. 92-101, 2001.
[8] J. Liu, I. Matta, M. Crovella, “End-to-End Inference of Loss Nature in a Hybrid Wired/Wireless Environment”, T.R., Boston Univ., Mar. 2002.
[9] W. Turin, M.M. Sondhi, “Modeling Error Sources in Digital Channels”, *IEEE Journal on Selected Areas on Communications*, Vol. 11(3), pp. 340-347, Apr. 1993.
[10] M. Zorzi, R.R. Rao, L.B. Milstein, “On the Accuracy of a First-Order Markov Model for Data Block Transmission on Fading Channels”, *IEEE ICUPC*, pp. 211-215, Nov. 1995.
[11] W. Wei, B. Wang, D. Towsley, J. Kurose, “Model-based Identification of Dominant Congested Links”, *ACM SIGCOMM Int. Measurement Conf.*, Oct 2003.
[12] S. Tao, R. Guérin, “On-line Estimation of Internet Path Performance: An Application Perspective”, *IEEE INFOCOM*, March 2004.
[13] P. Salvo Rossi, “End-to-End Packet-Channels Modeling”, PhD Thesis, Univ. of Naples “Federico II”, Jan. 2005.
[14] Y. Bengio, P. Frasconi, “Input-Output HMM’s for Sequence Processing”, *IEEE Trans. on Neural Networks*, Vol. 7(5), pp. 1231-1249, Sept. 1996.
[15] C.J. Bovy, H.T. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, P. Van Mieghem, “Analysis of End-to-End Delay Measurements in Internet”, *PAM Workshop*, pp. 26-33, Mar. 2002.
[16] L.R. Rabiner, “A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, *Procs. of the IEEE*, Vol. 77(2), pp. 257-285, Feb. 1989.
[17] D-ITG, Distributed Internet Traffic Generator. `http://www.grid.unina.it/software/ITG`
[18] N. Fonseca, M. Crovella, “Bayesian Packet Loss Detection for TCP”, *IEEE INFOCOM*, March 2005