Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Internet traffic modeling by means of Hidden Markov Models [☆,☆☆]

Alberto Dainotti [a], Antonio Pescapé [a], Pierluigi Salvo Rossi [b,c], Francesco Palmieri [c], Giorgio Ventre [a,*]

[a] Department of Computer Science and Systems, University of Naples "Federico II", Via Claudio 21, 80125 Napoli, Italy
[b] Department of Electronics and Telecommunications, Norwegian University of Science and Technology, O.S. Bragstads plass 2B, 7491 Trondheim, Norway
[c] Department of Information Engineering, Second University of Naples, Via Roma 29, 81031 Aversa (CE), Italy

ARTICLE INFO

Article history:
Received 1 March 2007
Received in revised form 12 December 2007
Accepted 7 May 2008
Available online 28 May 2008

Responsible Editor: M. Smirnow

ABSTRACT

In this work, we propose a Hidden Markov Model for Internet traffic sources at packet level, jointly analyzing Inter Packet Time and Packet Size. We give an analytical basis and the mathematical details regarding the model, and we test the flexibility of the proposed modeling approach with real traffic traces related to common Internet services with strong differences in terms of both applications/users and protocol behavior: SMTP, HTTP, a network game, and an instant messaging platform. The presented experimental analysis shows that, even maintaining a simple structure, the model is able to achieve good results in terms of estimation of statistical parameters and synthetic series generation, taking into account marginal distributions, mutual, and temporal dependencies. Moreover we show how, by exploiting such temporal dependencies, the model is able to perform short-term prediction by observing traffic from real sources.

1. Introduction

Understanding and solving performance-related issues of current and future networks requires the availability of realistic, but still simple and manageable, traffic models. Therefore the modeling of Internet traffic represents a critical task in the study and in the design of Internet architectures. Many efforts have focused on modeling source traffic related to specific application-level protocols, also with the purpose of conducting realistic network traffic simulation and emulation experiments (i.e. generating synthetic traffic in real networks).

Here we present a source-based modeling approach relying on a packet-level view of Internet traffic. The analysis of network traffic, indeed, can be made at different abstraction levels, e.g. session, conversation, connection/flow, packet, byte. With the term packet-level we mean the characterization of traffic in terms of *Inter Packet Time* (IPT) and *Packet Size* (PS). Such approach results particularly attractive because of its conciseness, flexibility, and because it allows to look at traffic from the lowest point of view. With the term source-based approach, we mean a traffic characterization and modeling of traffic generated by Internet applications running on single hosts.

As for the analytical approach, we adopted a specifically suited Hidden Markov Model (HMM). The idea is to keep the model analytically simple and tractable, but capable to capture important joint dynamics (in terms of both marginal distributions and time dependencies) of IPT and PS. We evaluate the model capabilities (*learning*, *generation*, and *prediction*) in order to construct realistic packet-level

[☆] This work has been partially supported by PRIN 2007 RECIPE Project, by CONTENT EU Network of Excellence, and finally from the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement No. 216585 (INTERSECTION Project).

[☆☆] Preliminary results within the same framework of this work have been recently published in [1,2].

[*] Corresponding author. Tel.: +39 081 7683908; fax: +39 081 7682950.
E-mail addresses: alberto@unina.it (A. Dainotti), pescape@unina.it (A. Pescapé), salvoros@iet.ntnu.no (P. Salvo Rossi), francesco.palmieri@unina2.it (F. Palmieri), giorgio@unina.it (G. Ventre).

1389-1286/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2008.05.004

models from the automated analysis of empirical traffic traces, by considering the marginal distributions and the auto- and cross-covariances of IPT and PS. In addiction, we compare synthetically generated sequences of IPT–PS pairs against those from real traces. Also, we show the capability of the proposed model to predict the short-term future behavior of the analyzed traffic on the basis of a very small amount of monitored traffic.

An important objective of this work is the design of a model flexible enough to work with different kinds of Internet traffic sources. For this reason, we apply our approach to traffic traces of various application-layer protocols and related to very different Internet services. More precisely we separately consider traffic generated by (i) SMTP, (ii) HTTP, (iii) a network game, and (iv) an instant messaging application. The conducted experimental investigation shows that, with a very limited complexity, the proposed model achieves acceptable results. Moreover, the prediction capabilities of the model – tested here in an off-line fashion – let foresee the useful application of such modeling approach for resource reservation and admission control purposes.

Finally, according to the source-based approach, we do not focus on aggregate link traffic, whereas we separately analyze several sessions of traffic exiting from single hosts related to specific application-level protocols. The purpose is to model the *average single session* for each considered application. The effect of superposition of multiple synthetic traffic sources, e.g. the presence of self-similarity and long range dependence in the synthetic aggregated traffic, falls beyond the scope of this paper.

The rest of the paper is organized as follows. In Section 2, related works and motivations at the basis of this work are given. Section 3 provides an introduction to HMM and a description on their application to build the proposed analytical model, furnishing details about model statistics and the learning stage. Section 4 describes the measurement approach, giving insights and motivations on the specific traffic taken in consideration. In Section 5, we show results of the model applied to SMTP, HTTP, Age of Mythology (AoM), and MSN Messenger. Section 6 ends the paper discussing the presented results and giving conclusion remarks.

2. Motivation and related work

Source traffic models are necessary to reproduce realistic user/application behavior in simulative environments or in network testbeds by injecting synthetic network traffic (e.g. traffic emulation). This allows to study network architectures performance problems by reconstructing the flows of packets generated by single sources. In the past, several source models related to HTTP traffic have been proposed [3,4], being the dominant Internet application, whereas only simple statistical characterizations of source traffic related to other applications like SMTP, network games, etc., have been presented [5,6]. Past years though, have seen a growing heterogeneity of Internet applications, making necessary the availability of models for different kinds of applications. Here, we explore the feasibility of a single modeling approach, flexible enough to work with different categories of sources, to be easily integrated into a traffic generation (or simulation) framework [7]. Moreover, even if we stress that this was the main focus of the present work, as regards some of the considered traffic categories, as network games, we would like to notice that this work represents one of the first attempts to build a thorough statistical model able to take into account multiple properties of the traffic. Indeed, while there exists a rich literature in terms of traffic characterizations and sometimes modeling of network games [6,8–11], this usually focuses on fitting the marginal distributions of IPT and PS, sometimes by arbitrarily splitting the fitting into different analytical distributions for different portions of the sample set; time dependence (a relevant property to consider when studying traffic modeling and simulation) and mutual dependence are usually not taken into account.

The proposed model relies on HMMs to reproduce traffic sources at packet-level. The reason for we focused on a packet-level view of traffic is that it provides the following benefits when compared with higher-level approaches: (i) we look at traffic at the deepest level of detail but at the same time basing the observations on just two variables; (ii) switching devices often operate on a packet-by-packet basis, therefore it is important to dispose of realistic packet-level models to evaluate their performance; (iii) most network performance problems (e.g. Loss, Delay, Jitter) happen at packet level; (iv) working at packet-level makes our approach independent of protocols evolution and applicable to different applications/protocols; (v) such kind of model is usable in traffic generators and simulators; (vi) traffic at packet level remains observable after encryption made by, for example, end-to-end cryptographic protocols such as SSL or IPSec; (vii) packet-level traffic models make robust approaches to traffic profiling for anomaly detection.

As far as concerns the analytical modeling approach, we had to face the trade-off among accuracy (the capability to capture as much statistical properties of traffic dynamics as possible), flexibility, and simplicity. The use of HMMs allowed us to build an easily tractable model, capable to jointly take into account IPT and PS first order statistics as well as temporal dynamics and correlation. In spite the large number of references related to network traffic modeling, very few works aim at joint modeling of IPT and PS [12–14]. Whereas, it has been demonstrated that neglecting aspects related to PS (e.g. assuming a constant value) significantly affects performance analysis [12]. Correlation structure is also a fundamental aspect that must be considered [15] when realistic replication of traffic is needed.

Recently the interest in HMM-based models has grown, and HMM models have been proposed as a tool for several network traffic related research problems. In [16,17] HMM models have been used to model the states of packet channels via corresponding loss probabilities and end-to-end delay distributions. Similar works have been proposed to model wired [18] and wireless [19] packet channels. To the best of our knowledge, few modeling works using HMMs to model traffic sources at packet level are present in literature. Specifically, we found approaches to Internet traffic modeling able to capture temporal structures based on MMPP (Markov Modulated Poisson Process) [20] and BMAP (Batch Markovian Arrival Process) [12,13]. In [20], authors propose a layered model to replicate traffic at edge routers that takes into account hierarchical characteristics of Internet traffic as well as long-range dependence properties, while in [13] efficient implementation of analytical tractable models for aggregate IP traffic is presented focusing on burstiness and self-similarity properties. The BMAP model proposed in [12], which considers both packet IPT and PS, is designed to capture the long-range dependence present in traffic traces of aggregate link traffic and it is evaluated in terms of queue analysis. In our work, instead, we concentrate on the traffic generated by single sources which is then mixed in network links, therefore here we do not consider queue analysis. In [21], a Markov-based model has been proposed and applied to variable bitrate MPEG traffic; GOP layer traffic characteristics for MPEG video traffic and sources are constructed from MPEG1 encoded video sequences. The same model has been applied to other traffic types (e.g, VoIP traffic). In [22], HMMs have been used to disjointly model IPT and PS of both aggregated and WWW traffic, comparing results against those from a stochastic generator based on a chaotic attractor. Finally, in [23], again considering IPT and PS disjointly, HMMs have been used to build traffic classifiers based on packet-level statistics related to some Internet applications. It is worth noticing that the HMM-based modeling approach presented here is part of a more general framework that also includes packet-channels modeling [18,19]. The long-term objective is a powerful homogeneous analytical framework for effective modeling of packet-level environments in heterogeneous scenarios (both in terms of traffic sources and end-to-end network paths).

To highlight and summarize the significance of the approach proposed in this work, we underline that, to the best of our knowledge, it extends the results present in literature in that

- it allows IPT/PS joint description;
- it allows synthetic series generation of both IPT and PS;
- it allows source state estimation with traffic prediction;
- it is derived by real traffic traces;
- it has been tested on different traffic types (quite different from each other in terms of both used protocols and users/applications behavior), deriving analogies and differences on the equivalent traffic models;
- results obtained with the analyzed traffic categories show the flexibility of the proposed model making it generalizable;
- as regards games traffic, this represents one of the first works to present a more complete model taking into account the aforementioned statistical properties.

Finally, we would like to underline that at [24], we make publicly available the open-source tool (called *Plab*) used for traffic analysis and measurement at packet-level, the algorithms developed for the analytical model, and, finally, the large set of heterogeneous data/traffic traces used in this work.

3. The model

3.1. Hidden Markov Models

We propose a statistical model[1] for packet-level network traffic. More specifically, we model the single source of traffic as an HMM. Generally speaking, an HMM may be viewed as a probabilistic function of a (hidden) Markov chain [25], thus it is composed of 2 variables:

- the hidden-state variable, whose temporal evolution follows a Markov-chain behavior;
- the observable variable, that stochastically depends on the hidden state.

Its topology is shown in Fig. 1, where $x_n \in \{s_1, \ldots, s_N\}$ and $y_n \in \{o_1, \ldots, o_M\}$ represent the state and the observable at discrete time n, respectively, with N and M being the number of states and the number of observable, respectively. An HMM is characterized by

- $\mathbf{u}$ – the initial state distribution, where $u_i = \Pr(x_1 = s_i)$;
- $\mathbf{A}$ – the $N \times N$ state transition matrix, where $A_{i,j} = \Pr(x_n = s_j | x_{n-1} = s_i)$;
- $\mathbf{B}$ – the $N \times M$ observable generation matrix,[2] where $B_{i,j} = \Pr(y_n = o_j | x_n = s_i)$.

We denote $\lambda = \{\mathbf{u}, \mathbf{A}, \mathbf{B}\}$ the complete set of parameters. The three fundamental problems of an HMM are

- *evaluation* – given a model λ and a sequence of observations $\mathbf{y} = (y_1, \ldots, y_L)$, compute efficiently the probability of the sequence given the model, $\Pr(\mathbf{y}|\lambda)$. It is solved via the forward–backward algorithm.
- *reconstruction* – given a model λ and a sequence of observations $\mathbf{y} = (y_1, \ldots, y_L)$, find the most likely corresponding sequence of states $\mathbf{x} = (x_1, \ldots, x_L)$. It is solved via the Viterbi algorithm, a dynamic programming technique performing computation of the *best score* and *tracking* variables.
- *learning* – given a sequence of observations $\mathbf{y} = (y_1, \ldots, y_L)$, find the set of parameters λ such that the likelihood of the model $\mathscr{L}(\mathbf{y}; \lambda) = \Pr(\mathbf{y}|\lambda)$ is maximum. It is solved via the Baum–Welch algorithm, a special case of the Expectation–Maximization algorithm [26], that iteratively updates the parameters in order to find a local maximum point of the parameter set.

It is worth noticing that the recursive computation of the *forward* and *backward* variables presents a complexity $o(LN^2)$ with respect to the complexity $o(LN^L)$ of direct calculation, with L being the length of the sequence of

[1] *Notation* – Upper (resp. lower) bold case letters denote matrices (resp. column vectors), $A_{i,j}$ (resp. a_i) denotes the (i,j)th (resp. ith) element of matrix $\mathbf{A}$ (resp. column vector $\mathbf{a}$), $\mathbf{1}$ denotes a column vector whose elements are 1, $\delta_{i,j}$ denotes the delta of Kronecker, $[\cdot]^{\mathrm{T}}$ and $\mathbb{E}\{\cdot\}$, respectively, denote transpose and expectation operators, the symbol $\sim$ means "distributed as".

[2] If the observable variable is continuous, the observable matrix is replaced with a set of N conditional pdfs, say $\{B_1(y), B_2(y), \ldots, B_N(y)\}$.

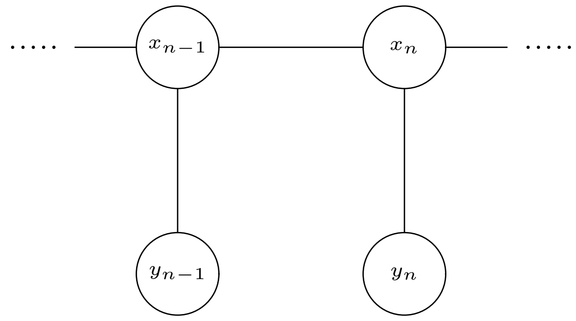


Fig. 1. Hidden Markov Model topology.

observations. For a more comprehensive discussion on HMMs refer to [25,27,28].

3.2. A packet-level source model

Referring to a single source of traffic, we consider an HMM in which the state variable is discrete, $x_n \in \{s_1, \ldots, s_N\}$, and the observable variable is a continuous bi-dimensional vector, $\mathbf{y}_n = [d_n, b_n]^{\mathrm{T}}$. The first and second components of $\mathbf{y}_n$ represent the IPT and the PS for the nth packet, respectively in dBμ (which we define as $10\log_{10}(\mathrm{IPT}/1\ \mu\mathrm{s})$) and in bytes.[3] The state variable has been introduced to account for memory and correlation phenomena between IPT and PS. We assumed that IPT and PS are statistically independent given the state. Also, in order to reduce the number of parameters, we assume $\mathbf{u} = \mathbf{q}$, where $\mathbf{q}$ is the steady-state distribution,[4] given by $\mathbf{A}^{\mathrm{T}}\mathbf{q} = \mathbf{q}$.

$\Lambda = \{\mathbf{A}, \mathbf{g}^{(t)}, \mathbf{w}^{(t)}, \mathbf{g}^{(p)}, \mathbf{w}^{(p)}\}$ is the set of parameters characterizing the model, denoting the state transition matrix, the conditional IPT and PS distribution vectors, respectively, i.e.

- $A_{i,j} = \Pr(x_{n+1} = s_j | x_n = s_i)$;
- $d_n | x_n = s_i \sim \Gamma(g_i^{(t)}, w_i^{(t)})$;
- $b_n | x_n = s_i \sim \Gamma(g_i^{(p)}, w_i^{(p)})$;

then the conditional pdfs for IPT and PS are:

$$f_i^{(t)}(d) = \frac{(d/w_i^{(t)})^{g_i^{(t)}-1}\,\mathrm{e}^{-(d/w_i^{(t)})}}{w_i^{(t)}\Gamma(g_i^{(t)})}\,(d > 0),$$

$$f_i^{(p)}(b) = \frac{(b/w_i^{(p)})^{g_i^{(p)}-1}\,\mathrm{e}^{-(b/w_i^{(p)})}}{w_i^{(p)}\Gamma(g_i^{(p)})}\,(b > 0).$$

The choice of Gamma distributions for IPT and PS is because a mixture of normal distributions can easily approximate a general distribution, Gamma is practically very similar to a normal distribution and has the desirable characteristic to be null for negative values (being negative IPT and PS meaningless). Summarizing we have a model where x_n is a discrete random variable whose dynamic behavior is governed by the transition matrix $\mathbf{A}$, with a Markovian assumption for the evolution, and $\mathbf{y}_n$ is a bi-dimensional continuous random variable describing IPT and PS as mixtures of conditionally independent (given the state) Gamma distributions, i.e.

$$f_i(\mathbf{y}_n) = f_i^{(t)}(d_n) f_i^{(p)}(b_n). \tag{1}$$

3.2.1. Model statistics

The IPT and PS conditional means and standard deviations are

$$\mu_i^{(t)} = g_i^{(t)} w_i^{(t)}, \quad \sigma_i^{(t)} = \sqrt{g_i^{(t)}}\,w_i^{(t)}, \quad \mu_i^{(p)} = g_i^{(p)} w_i^{(p)},$$
$$\sigma_i^{(p)} = \sqrt{g_i^{(p)}}\,w_i^{(p)}, \tag{2}$$

respectively, due to the conditional Gamma distribution assumption, then the IPT and PS global means and standard deviations of the model are

$$\mu^{(t)} = \sum_{i=1}^{N} q_i \mu_i^{(t)}, \quad \sigma^{(t)} = \sqrt{\sum_{i=1}^{N} q_i \mu_i^{(t)} (1 + g_i^{(t)}) w_i^{(t)} - (\mu^{(t)})^2},$$
$$\mu^{(p)} = \sum_{i=1}^{N} q_i \mu_i^{(p)}, \quad \sigma^{(p)} = \sqrt{\sum_{i=1}^{N} q_i \mu_i^{(p)} (1 + g_i^{(p)}) w_i^{(p)} - (\mu^{(p)})^2}. \tag{3}$$

Also, global IPT and PS pdfs are

$$f_{\mathrm{IPT}}(d) = \sum_{i=1}^{N} q_i f_i^{(t)}(d), \quad f_{\mathrm{PS}}(b) = \sum_{i=1}^{N} q_i f_i^{(p)}(b).$$

The conditional (given that state) duration in the state s_i is

$$\phi_i = \frac{1}{1 - A_{i,i}}. \tag{4}$$

IPT and PS auto- and cross-correlations of the model are

$$R^{(t)}(m) = \mathbb{E}\{d_n d_{n+m}\} = \begin{cases} \mathbf{q}^{\mathrm{T}} \mathbf{E}_{\mathrm{II}}^{(t)} \mathbf{1}, & m = 0, \\ \mathbf{q}^{\mathrm{T}} \mathbf{E}^{(t)} \mathbf{A}^{|m|-1} \mathbf{E}^{(t)} \mathbf{1}, & m \neq 0, \end{cases}$$

$$R^{(p)}(m) = \mathbb{E}\{b_n b_{n+m}\} = \begin{cases} \mathbf{q}^{\mathrm{T}} \mathbf{E}_{\mathrm{II}}^{(p)} \mathbf{1}, & m = 0, \\ \mathbf{q}^{\mathrm{T}} \mathbf{E}^{(p)} \mathbf{A}^{|m|-1} \mathbf{E}^{(p)} \mathbf{1}, & m \neq 0. \end{cases}$$

$$R^{(tp)}(m) = \mathbb{E}\{d_n b_{n+m}\} = \begin{cases} \mathbf{q}^{\mathrm{T}} \mathbf{E}_{\mathrm{II}}^{(tp)} \mathbf{1}, & m = 0, \\ \mathbf{q}^{\mathrm{T}} \mathbf{E}^{(t)} \mathbf{A}^{|m|-1} \mathbf{E}^{(p)} \mathbf{1}, & m \neq 0. \end{cases}$$

where

$$E_{\mathrm{II}i,j}^{(t)} = A_{i,i}(1 + g_i^{(t)}) g_i^{(t)} (w_i^{(t)})^2 \delta_{i,j}, \quad E_{i,j}^{(t)} = A_{i,i} g_i^{(t)} w_i^{(t)} \delta_{i,j},$$
$$E_{\mathrm{II}i,j}^{(p)} = A_{i,i}(1 + g_i^{(p)}) g_i^{(p)} (w_i^{(p)})^2 \delta_{i,j}, \quad E_{i,j}^{(p)} = A_{i,i} g_i^{(p)} w_i^{(p)} \delta_{i,j},$$
$$E_{\mathrm{II}i,j}^{(tp)} = A_{i,i} g_i^{(t)} w_i^{(t)} g_i^{(p)} w_i^{(p)} \delta_{i,j}.$$

It is worth noticing that $\mathbf{E}$ and $\mathbf{E}_{\mathrm{II}}$ are first-order and second-order statistics matrices. To show traffic dynamics without the biasing effects of IPT and PS global means, in Section 5 covariances are taken into account instead of correlations.

[3] We measure IPT with a resolution of 1 μs (as explained in Section 4) and apply a logarithmic transformation because they range over several orders of magnitude.

[4] If x_n is an irreducible and aperiodic process, the steady-state distribution equals the limit distribution, $q_i = \lim_{n\to\infty}\{\Pr(x_n = s_i)\}$, see [29].

3.2.2. Learning the model parameters

The Expectation–Maximization algorithm [26] 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's this optimization technique reduces to the Baum–Welch algorithm [25,27,28], studied for discrete and continuous observable variables with a broad class of allowed conditional pdfs. The Baum–Welch algorithm is an iterative procedure looking for a local maximum of the likelihood function which typically depends on the starting point Λ. When necessary, multiple trainings with different initial conditions provide the global solution.

More specifically, consider a set of observable sequences $\mathbf{Y} = \{\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(K)}\}$ referred to as the *training set*, where each sequence $\mathbf{Y}^{(k)} = [\mathbf{y}_1^{(k)}, \ldots, \mathbf{y}_{L_k}^{(k)}]$ represents IPT and PS from a single session.[5] We want to find the set of parameters such that the likelihood $\mathscr{L}(\mathbf{Y}; \Lambda) = \Pr(\mathbf{Y}|\Lambda)$ of the training set is maximum. The Baum–Welch for the proposed source-traffic model is then based on the following equations:

$$\hat{A}_{i,j} = \frac{\sum_{k=1}^{K} \frac{1}{\mathscr{L}^{(k)}} \sum_{n=1}^{L_k-1} \alpha_n^{(k)}(i) A_{i,j} f_j(\mathbf{y}_{n+1}^{(k)}) \beta_{n+1}^{(k)}(j)}{\sum_{k=1}^{K} \frac{1}{\mathscr{L}^{(k)}} \sum_{n=1}^{L_k-1} \alpha_n^{(k)}(i) \beta_n^{(k)}(i)},$$

$$\hat{g}_i^{(t)} \hat{w}_i^{(t)} = \frac{\sum_{k=1}^{K} \frac{1}{\mathscr{L}^{(k)}} \sum_{n=1}^{L_k} \alpha_n^{(k)}(i) \beta_n^{(k)}(i) d_n^{(k)}}{\sum_{k=1}^{K} \frac{1}{\mathscr{L}^{(k)}} \sum_{n=1}^{L_k-1} \alpha_n^{(k)}(i) \beta_n^{(k)}(i)},$$

$$\hat{g}_i^{(p)} \hat{w}_i^{(p)} = \frac{\sum_{k=1}^{K} \frac{1}{\mathscr{L}^{(k)}} \sum_{n=1}^{L_k} \alpha_n^{(k)}(i) \beta_n^{(k)}(i) b_n^{(k)}}{\sum_{k=1}^{K} \frac{1}{\mathscr{L}^{(k)}} \sum_{n=1}^{L_k-1} \alpha_n^{(k)}(i) \beta_n^{(k)}(i)},$$

$$\hat{g}_i^{(t)} (\hat{w}_i^{(t)})^2 = \frac{\sum_{k=1}^{K} \frac{1}{\mathscr{L}^{(k)}} \sum_{n=1}^{L_k} \alpha_n^{(k)}(i) \beta_n^{(k)}(i) (d_n^{(k)} - \mu_i^{(t)})^2}{\sum_{k=1}^{K} \frac{1}{\mathscr{L}^{(k)}} \sum_{n=1}^{L_k-1} \alpha_n^{(k)}(i) \beta_n^{(k)}(i)},$$

$$\hat{g}_i^{(p)} (\hat{w}_i^{(p)})^2 = \frac{\sum_{k=1}^{K} \frac{1}{\mathscr{L}^{(k)}} \sum_{n=1}^{L_k} \alpha_n^{(k)}(i) \beta_n^{(k)}(i) (b_n^{(k)} - \mu_i^{(p)})^2}{\sum_{k=1}^{K} \frac{1}{\mathscr{L}^{(k)}} \sum_{n=1}^{L_k-1} \alpha_n^{(k)}(i) \beta_n^{(k)}(i)},$$

where referring to the kth sequence, the likelihood is

$$\mathscr{L}^{(k)} = \Pr(\mathbf{Y}^{(k)}|\Lambda) = \sum_{i=1}^{N} \alpha_n^{(k)}(i) \beta_n^{(k)}(i),$$

and the Forward and Backward variables are computed according to the following recursions:

$$\alpha_n^{(k)}(j) = \begin{cases} \sum\limits_{i=0}^{N-1} \alpha_{n-1}^{(k)}(i) A_{i,j} f_j(\mathbf{y}_n^{(k)}), & n = 1, \ldots, L_k, \\ \delta_{1,j}, & n = 0, \end{cases}$$

$$\beta_n^{(k)}(i) = \begin{cases} \sum\limits_{j=0}^{N-1} A_{i,j} f_j(\mathbf{y}_{n+1}^{(k)}) \beta_{n+1}^{(k)}(j), & n = 0, \ldots, L_k - 1, \\ 1, & n = L_k. \end{cases}$$

In our experiments the initialization for the parameter set Λ, has been such to have the conditional pdfs, for both IPT and PS, uniformly distributed on the whole observed range. More specifically, the state-transition matrix is given by

$$A_{i,j} = 1/N, \tag{5}$$

while denoting

$$d_{\min} = \min_{k,n} d_n^{(k)}, \quad d_{\max} = \max_{k,n} d_n^{(k)},$$
$$b_{\min} = \min_{k,n} b_n^{(k)}, \quad b_{\max} = \max_{k,n} b_n^{(k)},$$

then $\{\mathbf{g}^{(t)}, \mathbf{w}^{(t)}, \mathbf{g}^{(p)}, \mathbf{w}^{(p)}\}$ are chosen as

$$\begin{aligned} \mu_{i+1}^{(t)} - \mu_i^{(t)} &= \frac{d_{\max} - d_{\min}}{N+1}, \quad \sigma_i^{(t)} = \frac{d_{\max} - d_{\min}}{5(N+1)}, \\ \mu_{i+1}^{(p)} - \mu_i^{(p)} &= \frac{b_{\max} - b_{\min}}{N+1}, \quad \sigma_i^{(p)} = \frac{b_{\max} - b_{\min}}{5(N+1)}. \end{aligned} \tag{6}$$

4. Traffic traces and measurement approach

4.1. Considered traffic

To verify its flexibility and general applicability, the proposed modeling approach has been tested with different categories of Internet traffic sources. The choice of such applications takes into account the level of novelty and popularity. Also, we considered applications differing from several points of views which all reflect into traffic peculiarities: man–computer interaction, transferred objects, underlying network protocols, etc.

The list of considered Internet applications, along with details of the corresponding traffic traces we used,[6] is reported in Table 1.

Firstly, we considered more traditional services as the Web and the Email. Although HTTP and SMTP are two applications largely involving all the Internet population (the most used by common users), they substantially differ for the kinds of treated objects as well as the level of user interaction. The characteristics of traffic generated by HTTP clients can be heavily affected by the human factor, above all as regards timings [2,3], whereas SMTP clients traffic is affected by users mostly in terms of the number and size of packets to be transferred.

Secondly, we considered applications which have become popular in the recent years and currently represent an increasing portion of the overall Internet traffic: instant messaging and multi-player network games. They both present novel and interesting characteristics with respect to other applications. Due to these differences, as for both games and instant messaging, the interest in the characterization and modeling of their traffic is increased in the last years [6,11,31,32].

Network games have strict latency requirements and traffic properties which substantially differ from more traditional Internet applications [6]. Moreover, while their traffic represents a relevant percentage yet – in [33] it was reported that about 4% of all packets in a backbone could be associated with only six popular network games – it is constantly increasing. Thus, analysis of such traffic is crucial to properly design and provision networks for future needs. We studied traffic generated by Age of Mythology (AoM), a Microsoft Real Time Strategy Multiplayer

[5] The meaning of "session" will be better defined in Section 4.

[6] Apart the AoM traffic traces available at [30], they are freely available at [24].

Table 1
Traffic traces details

Traffic	Link	Protocol	Port	Date	Size	Pkts	Sessions
SMTP	WAN	TCP	25	9/2005	3 GB	43 M	56 K
HTTP	WAN	TCP	80	7/2004	60 GB	830 M	1 M
AoM	LAN	UDP	2300	8/2003	12 MB	180 K	6
MSN	WAN	TCP	1863	4/2006	1 GB	9 M	1 M

Game [34]. As regards Instant Messengers, they are used by 50% of the Internet users all around the world [35], being MSN Messenger the most popular application, followed by AOL and Yahoo Messenger. In this work we model the traffic generated by MSN Messenger (MSN in the following) clients [36]. The level of user interaction in these kind of applications is obviously much higher. Moreover, because they represent a new vehicle of viruses, worms, and of other kinds of malicious use, the study of instant messaging applications, besides email, has also interesting security implications. For example, the traffic behavior characterization and modeling of such applications could be exploited for security purposes (classification, detection, prevention).

In Table 1 details about the traffic traces that we analyzed are given. As regards SMTP, HTTP, and MSN, we captured traffic by passively monitoring the WAN access link at *University of Napoli "Federico II"* network during the period *January 2004–April 2006*. The observed link represents the only connection of the University network to the Internet, and it has a maximum throughput equal to 200 Mbps.

With the term "*session*", in the case of SMTP (resp. HTTP), we mean all the traffic exchanged between two hosts related to port TCP 25 (resp. TCP 80), with a timeout of 15 min. As regards SMTP, we present results from the sessions with less than 100 packets, which we defined as *short-lived*, and which account for $\sim 97\%$ of the SMTP sessions. This is because we found that there are other sessions which exhibit extremely different statistical properties. This was confirmed by a K-means clustering we performed using a few features per session, e.g. number of packets, bytes, IPT and PS mean and variance. Note that considering only this class does not affect our approach, as we do not want to provide a comprehensive model for SMTP traffic. At this stage we want to show the applicability of the proposed approach also to this kind of traffic. As regards MSN, the MSN protocol, uses a client–server communication model in which user clients interact with Microsoft servers that belong to the MSN Messenger network and which accept connections on TCP port 1863 [37]. There are mainly two kinds of servers, which offer services of *presence* and *instant messaging*, respectively, [38]. Analysis of both communication protocol and real traffic traces allowed us to identify the subnets associated to each service. We collected traffic related to both services and both directions (inbound and outbound). In this work we report results related only to the outbound direction (i.e. from the clients to the servers) of *instant messaging* traffic. Each session is made of all the traffic exchanged between a single client–server pair (related to server port TCP 1863), with an inactivity timeout of 15 min. The AoM traces, instead, have been provided, in Tcpdump format, by the Worcester Polytechnic Institute (WPI), MA (USA) [30]. They consist of packet sequences of complete gaming sessions, between two players, captured in a LAN environment. We consider an AoM *session* given by all the traffic exchanged from the beginning to the end of a match. Only six gaming sessions were studied, because packet-level traffic of RTS games has been demonstrated being very predictable and strongly dependent from the specific game application whereas it is poorly dependent from user behavior [39]. Indeed, past works studying the statistical characterization of the traffic generated by this game have used only such traces. Such works show that this traffic is substantially different from traffic of more classical network applications. Moreover, in [40] we showed results and commented regarding the invariance of gaming traffic when observed under different situations, which makes reasonable the use of a small number of traces. As regards SMTP, HTTP, MSN traces, instead, we observed a much larger set of sessions. This is because of the more complex nature of such traffic [2] and also because we could gather our own traces.

4.2. Tools and issues with the data

Obtaining and making available traffic data useful for characterization and modeling is a complex task, which not only consists into traffic collection and selection of the appropriate traffic flows, but it also involves activities such as data sanitization and anonymization (see Fig. 2).

We used Plab [2,24] to capture the traffic traces we collected and analyzed. Plab is an open-source software, partially based on the Libpcap library [41], that we developed for the analysis of live traffic and of file traces in *tcpdump* format, and focused on packet-level measurement and analysis. This platform, employed also in previous works on traffic analysis and modeling, is capable to efficiently analyze very large traffic traces and to separate traffic into different sessions. Depending on user-specified parame-

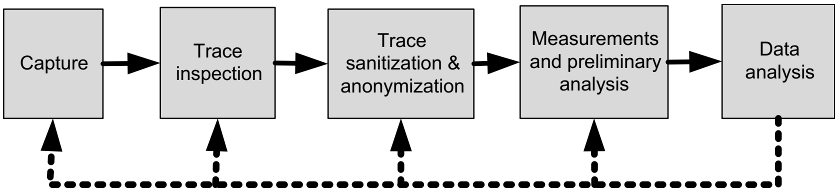

Capture
Trace inspection
Trace sanitization & anonymization
Measurements and preliminary analysis
Data analysis

Fig. 2. Life cycle of data analysis.

ters, a session can be identified by: (i) all packets sent and received by a host (*host mode*); (ii) all packets identified by source and destination IP and ports with a default timeout of 60 s (*flow mode*); (iii) all packets exchanged by 2 hosts related to a specific service (e.g. TCP port 80), with a user definable timeout (*conversation mode*). Given one of the above modes, sessions are assigned an ID, and for each session IPTs between packets flowing in the same direction are calculated, along with PS. We call such data packet-level data series.

With Plab it is possible to specify command line filters in *tcpdump/Berkeley Packet Filter* syntax to select the type of traffic to be captured or analyzed, e.g. layer 3 protocol, port, etc. Also, more intelligence was introduced into the software, as the ability to decode optional TCP headers like the MSS, or to filter packets or entire sessions based on several others criteria. For example, we introduced some payload inspection capabilities which served for data sanitization.

As regards sanitization, indeed, here we report on how we removed, from the considered data, samples related to traffic which was not HTTP, but tried to masquerade as it by running on TCP port 80. We instructed Plab to analyze the first 3 bytes of payload data exchanged between each host pair of a conversation related to TCP port 80. Under normal conditions, such bytes should correspond to the *method* invoked by the client in a HTTP request. As reported in Table 2, we observed that almost 94% of the sessions started with a GET request, 4% with a POST request, etc. Only a small fraction of the sessions presented packets starting with a byte not corresponding to an alphabetic character. Inside this category, 99% of the conversations started with the byte 0xe3, the first byte exchanged by peers opening a communication session based on the eDonkey2000 protocol [42], used by eMule and eDonkey file-sharing applications. Also, 0.44% of the sessions were initiated by the host communicating from port TCP 80 (labeled as “downstream” in Table 2). Because our interest was in modeling traffic generated only by applications running over HTTP, we instructed Plab to recognize such sessions and to filter them out. By filtering our traces, we observed that 5.12% of the processed packets were discarded. Therefore, this non-HTTP traffic represents a not negligible portion of the captured traffic. As regards the number of filtered sessions, they account for about 0.7% of the total. This suggests that the filtered sessions tend to generate more packets than authentic HTTP ones. By comparing the results obtained with and without filtering such sessions, we observed that discarded traffic had a consistent impact in terms of payload size and inter-packet time. Comparisons of the obtained distributions for upstream traffic at the UNINA site are shown in Fig. 3.

Table 2
Payload inspection on the first packet opening a conversation related to TCP port 80

Conversation start	GET	POS	HEA	Downstream	0xe3	PRO
Percentage	93.94	4.23	0.7	0.44	0.27	0.2

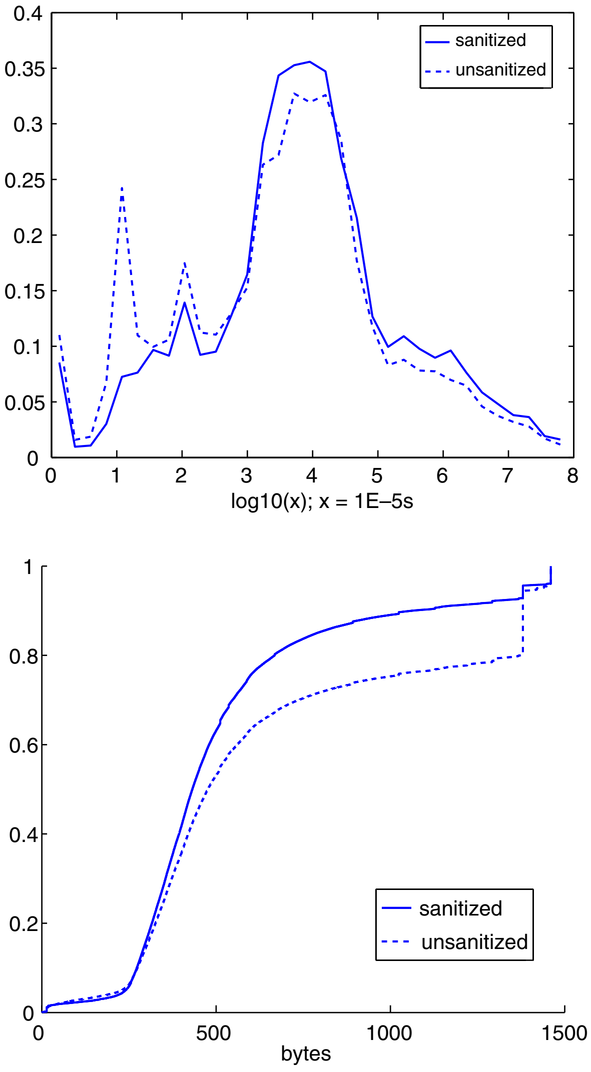


Fig. 3. Filtered UNINA upstream: IPT PDF (top), PS CDF (bottom).

Observing the properties of such distributions it is clear that the filtered sessions increase the portion of back-to-back packets with full payload, probably due to the presence of file-transfers. As reported in [2], after filtering out such traffic from the traces captured at the UNINA site, we found packet-level profiles strongly similar to those obtained by observing traffic at another site in which no Peer-to-Peer applications were running. This is not only a confirmation of the correct sanitization we performed, but also revealed important invariants (with respect to space and time) of the characteristics of the studied traffic.

The above example shows how acquiring realistic and reliable data to be used as a reference for traffic modeling is a delicate and sometimes not straightforward task, which requires attention and appropriate tools.

Finally, as regards data anonymization, to preserve users privacy we kept only the IP and TCP headers of each packet, and we scrambled IP addresses using the wide-tcpdpriv tool from the MAWI-WIDE project [43].

4.3. Measurement methodology and analyzed data

For each session between two hosts, depending on their direction, two separate flows of data can be identified, which we called *upstream* and *downstream*. In the case of SMTP, HTTP, and MSN, we identify as upstream the traffic flowing from a client to a server, that is, packets with destination port set to respectively TCP 80, TCP 25, and TCP 1863. Whereas downstream traffic is related to the opposite direction. In the case of SMTP, as regards downstream traffic, it is worth mentioning that the vast majority of downstream flows – for each session – are made of only few packets (about 5) of small size. Thus they represent a very small portion of SMTP traffic. This can explained by SMTP protocol specifications: the peer acting as a server usually answers to requests and data transfers from the client with small messages that must have a numeric ID prepended. As for HTTP instead, strong volumes of traffic are generated in both directions, this is due to the intrinsic nature of the Web traffic.

In this paper, we concentrate on the traffic sources represented by SMTP, HTTP, and MSN clients, we therefore model only upstream traffic. We adopt the same approach for AoM, modeling the traffic flowing in the outbound direction when seen from the point of view of a specific peer (i.e. leaving the workstation of a gaming user). Anyway, being the observed AoM traces related to matches with two players, the traffic flowing in the other direction is almost symmetrical.

An important aspect of our methodology is that in the evaluation of IPT and PS distributions we did not take into account packets with empty payload at transport level. Since we wanted to characterize the traffic generated by the applications, independently as much as possible of the transport level protocol itself, we decided to drop all TCP-specific traffic, like connection establishment packets (SYN-ACK-SYNACK) and pure acknowledgment packets [44]. For the same reason, in the estimation of the packet size, we measured the byte length of the TCP payload or, in the case of AoM we considered the UDP payload. These choices make our results usable for simulation purposes as an input for TCP state machines and UDP/IP stacks, like in D-ITG [24] and TCPlib [45].

As regards the time resolution of the measurements, the packet timestamping resolution provided by the Libpcap library (which is used both by Tcpdump and Plab), and by the kernel drivers that it links to, is of 1 μs. Moreover, because of the wide range of the considered IPTs, as reported in Section 3, we applied a logarithmic transformation to the measured values, $10\log_{10}(\mathrm{IPT}/1\ \mu\mathrm{s})$, which we will refer to as dBμ.

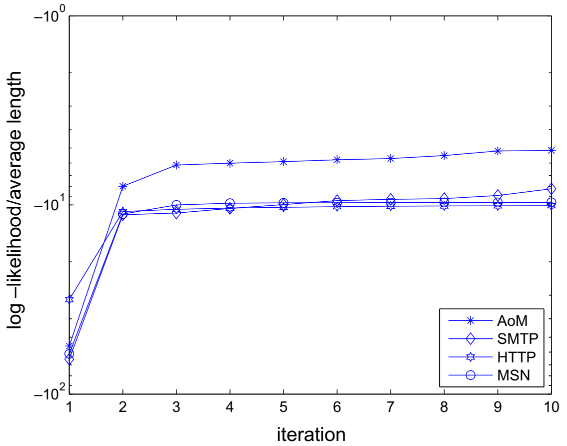


Fig. 4. Log-likelihood (normalized with respect to the average length of the session) vs. iteration.

Table 3
Model discrepancy: IPT-λ^2 and PS-λ^2

	IPT-λ^2	PS-λ^2
SMTP		
Starting	2.1×10^8	3.3×10^{46}
Trained	1.4	4.6
HTTP		
Starting	1.7×10^6	1.1×10^{44}
Trained	0.31	0.30
AoM		
Starting	0.99×10^2	2.2×10^{16}
Trained	0.24	1.6
MSN		
Starting	1.7×10^2	1.3×10^{47}
Trained	0.68	1.8

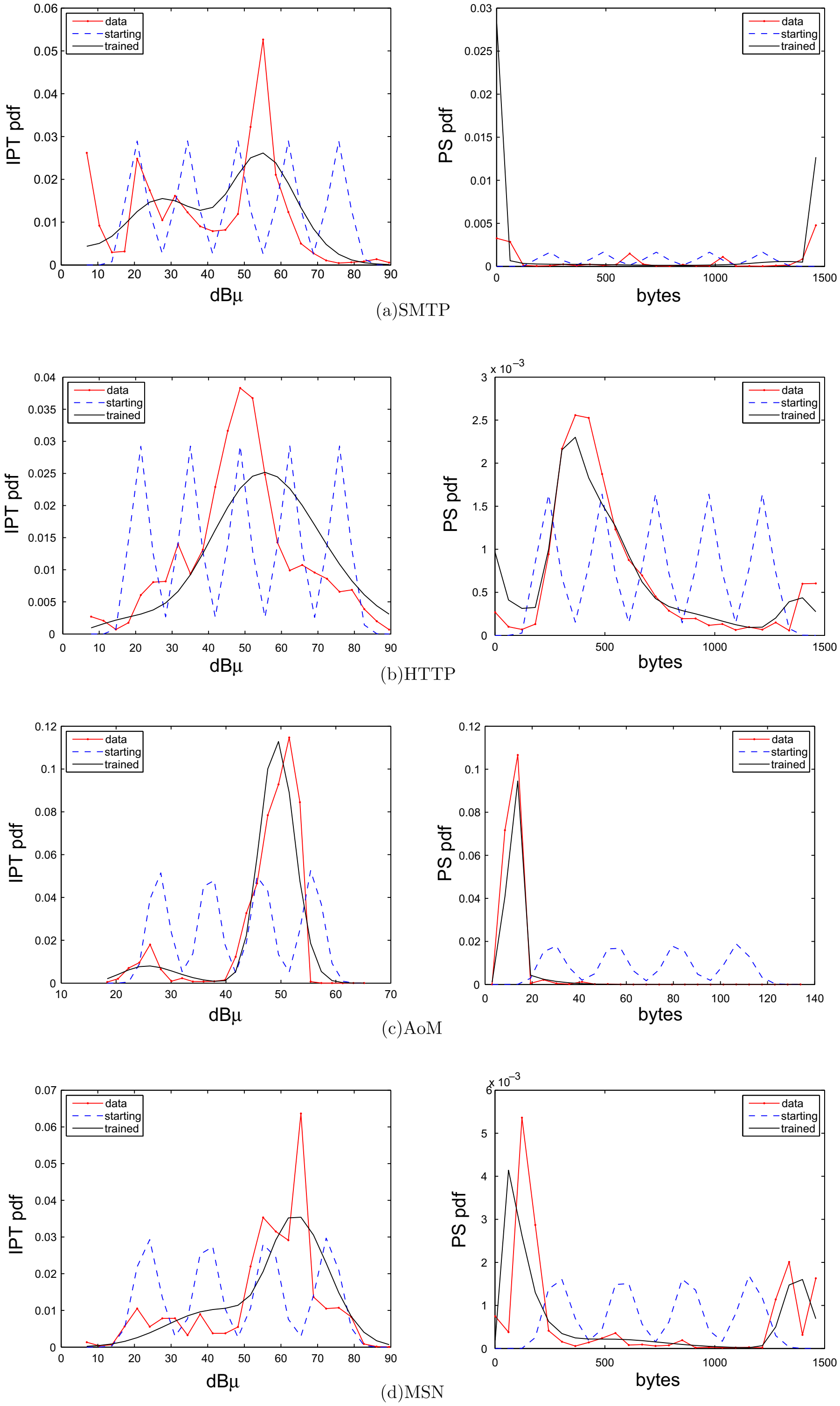
Table 4
Covariance EF: IPT-*K*, IPT-*v*, PS-*K*, PS-*v*, IPT/PS-*K* and IPT/PS-*v*

	IPT-*K*	IPT-*v*	PS-*K*	PS-*v*	IPT/PS-*K*	IPT/PS-*v*
SMTP						
Data	1.0	1.2	1.0	0.91	−0.31	0.46
Starting	1.0	50	1.0	50	1.0	50
Trained	1.0	0.43	1.0	0.25	−0.54	0.18
HTTP						
Data	1.0	0.75	1.0	0.63	0.16	1.1
Starting	1.0	50	1.0	50	1.0	50
Trained	1.0	1.8	1.0	0.29	0.17	0.98
AoM						
Data	1.0	42	1.0	42	−0.10	0.086
Starting	1.0	50	1.0	50	1.0	50
Trained	1.0	47	1.0	47	−0.17	2.4
MSN						
Data	1.0	1.0	1.0	0.49	−0.38	0.19
Starting	1.0	50	1.0	50	1.0	50
Trained	1.0	0.42	1.0	0.21	−0.59	0.17

5. Experimental results

This section presents some results of our model when it is applied to SMTP, HTTP, AoM, and MSN traffic.

We used the model with $N = 4$ states for AoM and MSN traffic and $N = 5$ states for SMTP and HTTP traffic, due to their more complex structure. Choices $N = 4$ and $N = 5$ have been found effective empirically, as they provided a sufficient number of modes to capture traffic behavior for the considered applications. In our experiments models with smaller number of states failed capturing the correct behavior, with some modes missing and/or correlation mismatches. Also, we do not explore models with larger number of states as increasing N provides a twofold negative effect: (i) increased computational complexity, affecting learning, monitoring, generation; (ii) more-likely "overfitting" problems, affecting prediction. It is worth



(a)SMTP

(b)HTTP

(c)AoM

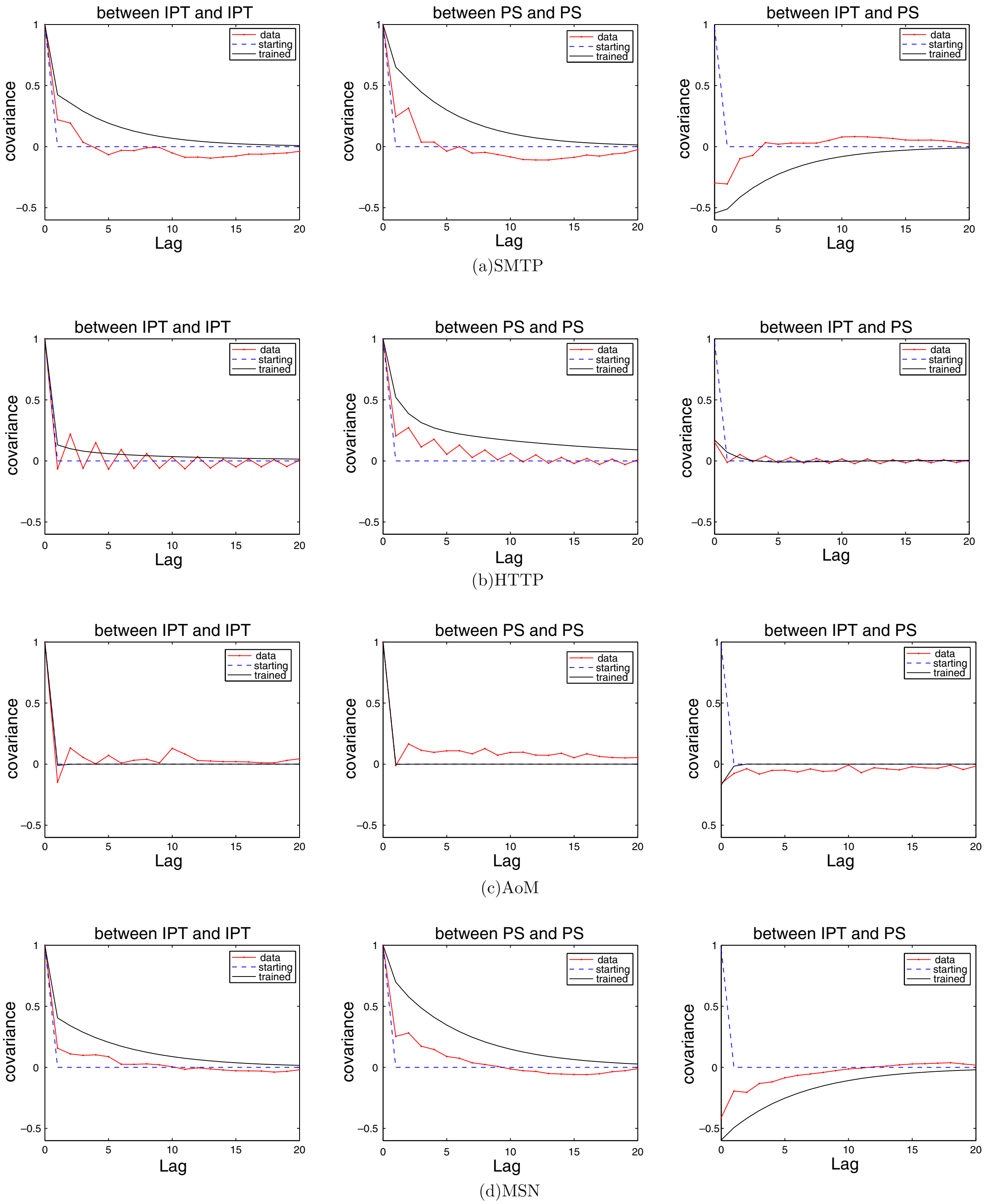
(d)MSN

Fig. 5. Histogram and pdf for IPT (left) and PS (right).

noticing that this section aims to provide: (i) the effectiveness of HMM's in traffic modeling at traffic level and (ii) specific HMM's for the 4 considered traffic typologies.

For all traffic typologies the learning algorithm converged in terms of likelihood after a few iterations. Fig. 4 shows the log-likelihood normalized with respect to the average length of the sessions. The reason behind normalization is because $\log(\mathscr{L}(\mathbf{Y};\Lambda))$ is decreasing with the length of the sequence $\mathbf{Y}$.

In the following, we denote "starting" model the uniformly distributed initialization for the parameters, while "trained" model, the parameters obtained after 10 itera-



(a)SMTP

(b)HTTP

(c)AoM

(d)MSN

Fig. 6. Auto- and cross-covariance for IPT and PS.

Table 5
Global statistics: traffic, model, global mean IPT (dBμ), global mean PS (bytes), IPT global st. dev. (dBμ), PS global st. dev. (bytes)

	$\mu^{(t)}$	$\mu^{(p)}$	$\sigma^{(t)}$	$\sigma^{(p)}$
SMTP				
Data	40	710	19	619
Starting	48	731	20	347
Trained	43	688	18	647
HTTP				
Data	49	542	15	324
Starting	49	731	19	347
Trained	56	541	17	348
AoM				
Data	47	12	8	4
Starting	42	69	11	30
Trained	47	12	8	4
MSN				
Data	56	557	16	570
Starting	48	731	19	331
Trained	58	511	15	561

tions of the Baum–Welch algorithm, the training set will simply be referred to as "data".

Table 3 compares the model discrepancy, for both the starting and the trained models, with respect to data via the parameter λ^2, commonly used in order to evaluate fitting distributions [48]. Table 4 compares the amplitude and decay parameters (K and v) for the covariances of data, starting model, and trained models when an exponential fitting (EF), with a minimum mean square error criterion, has been applied; that is, covariances are described in the form $K\exp(-v\text{lag})$. More specifically, the amplitude parameter is considered fixed ($K = 1$) for the auto-covariances, whereas it is a free parameter for the cross-covariances. Both tables show that the model, even if working in a jointly fashion, is able to fit with a good accuracy both marginal distributions and covariances. Figs. 5 and 6, analyzed in the following, will confirm graphically the results of both Tables 3 and 4. More specifically, Table 5 compares the global means and standard deviations for the starting and trained models with the data: the trained models exhibit good results in terms of mean and standard deviation for each considered traffic.

The starting values have been set via Eqs. (5) and (6) in Section 3, and are useful to show how, in few iterations, the model converges to values close to the empirical data. Such global statistics are obtained as weighted averages of each state conditional statistics, Eq. (3), whereas we will comment the behaviors of the single states in the following subsection, where different comparisons among data, starting and trained models, in terms of global pdfs (see Fig. 5), auto- and cross-covariance (see Fig. 6) are made.

Table 6
SMTP conditional statistics: state, steady-state probability, conditional mean IPT (dBμ), conditional mean PS (bytes), IPT conditional st. dev. (dBμ), PS conditional st. dev. (bytes), conditional duration

s_i	q_i	$\mu_i^{(t)}$	$\mu_i^{(p)}$	$\sigma_i^{(t)}$	$\sigma_i^{(p)}$	ϕ_i
s_1	0.447	57	48	8	134	10
s_2	0.109	18	542	14	269	1
s_3	0.176	31	1344	10	140	4
s_4	0.057	51	1107	7	204	1
s_5	0.211	35	1458	13	7	2

Table 7
HTTP conditional statistics: state, steady-state probability, conditional mean IPT (dBμ), conditional mean PS (bytes), IPT conditional st. dev. (dBμ), PS conditional st. dev. (bytes), conditional duration

s_i	q_i	$\mu_i^{(t)}$	$\mu_i^{(p)}$	$\sigma_i^{(t)}$	$\sigma_i^{(p)}$	ϕ_i
s_1	0.153	41	314	23	344	3
s_2	0.356	54	345	14	70	27
s_3	0.297	64	530	13	104	19
s_4	0.111	59	880	15	195	11
s_5	0.083	60	1387	15	77	3

Table 8
AoM conditional statistics: state, steady-state probability, conditional mean IPT (dBμ), conditional mean PS (bytes), IPT conditional st. dev. (dBμ), PS conditional st. dev. (bytes), conditional duration

s_i	q_i	$\mu_i^{(t)}$	$\mu_i^{(p)}$	$\sigma_i^{(t)}$	$\sigma_i^{(p)}$	ϕ_i
s_1	0.881	49	12	3	2	8
s_2	0.100	27	15	5	7	1
s_3	0.013	48	31	3	9	1
s_4	0.006	52	25	1	8	1

Table 9
MSN conditional statistics: state, steady-state probability, conditional mean IPT (dBμ), conditional mean PS (bytes), IPT conditional st. dev. (dBμ), PS conditional st. dev. (bytes), conditional duration

s_i	q_i	$\mu_i^{(t)}$	$\mu_i^{(p)}$	$\sigma_i^{(t)}$	$\sigma_i^{(p)}$	ϕ_i
s_1	0.578	66	104	8	73	7
s_2	0.092	44	437	17	299	1
s_3	0.060	61	655	5	193	1
s_4	0.270	45	1378	14	62	4

Some more considerations on the single modes discovered by the model are made looking at the conditional statistics (see Tables 6–9). In addition we made the trained models generate output variables to compare the synthetically generated IPT–PS pairs with those from real data (see Fig. 7). Finally the trained models are investigated in terms of prediction capabilities (see Figs. 10, 8, 9, 11).

5.1. Model construction and validation

5.1.1. SMTP traffic

Fig. 5 shows how SMTP traffic presents two main modes in the IPT distribution, separated by three orders of magnitude, and two main modes in the PS distribution, essentially made of very small packets and full payload packets respectively. More specifically, it is apparent from Table 6 that s_1 is responsible for transmission with large IPT and small PS, while s_3 and s_5 are responsible for transmission with small IPT and large PS. These three dominant modes account approximately for 85% (from steady state distributions) of the behavior of SMTP that mainly alternates two phases: s_1 with low bitrate and s_3 and s_5 with high bitrate. States s_2 and s_4 can be viewed as transient states to switch between these two modes.

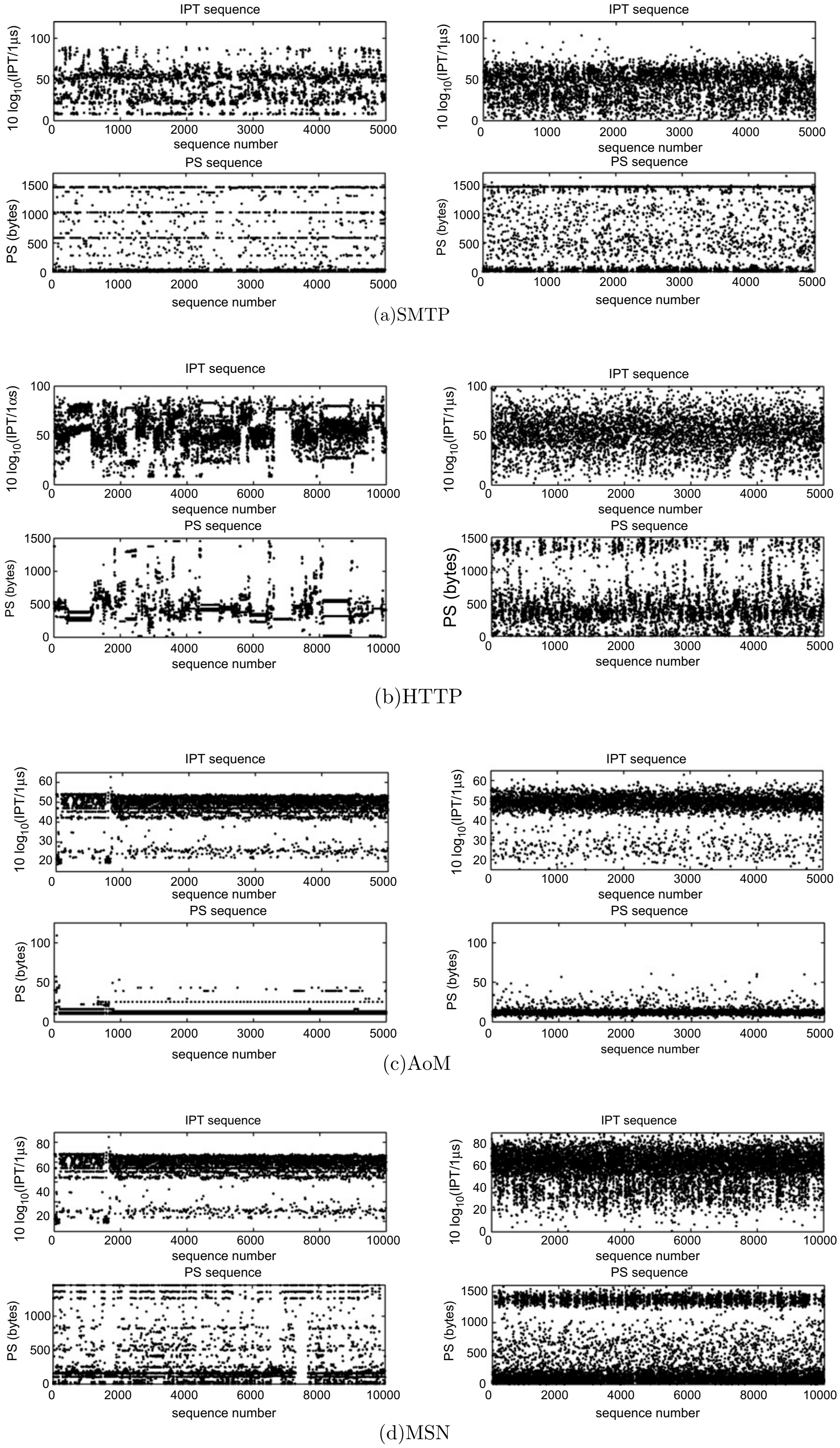


Fig. 7. Training (left) and Synthetic (right) traces for IPT and PS.

Fig. 6 shows how the model captures temporal dynamics, showing how both IPT and PS have a significant memory, due to the non negligible auto-covariances. More interesting is the negative cross-covariance, confirming that IPT and PS usually are not at the same time both large or small.

In Fig. 7 we show the capability of the model to jointly reproduce time series of both IPT and PS by synthetic gen-

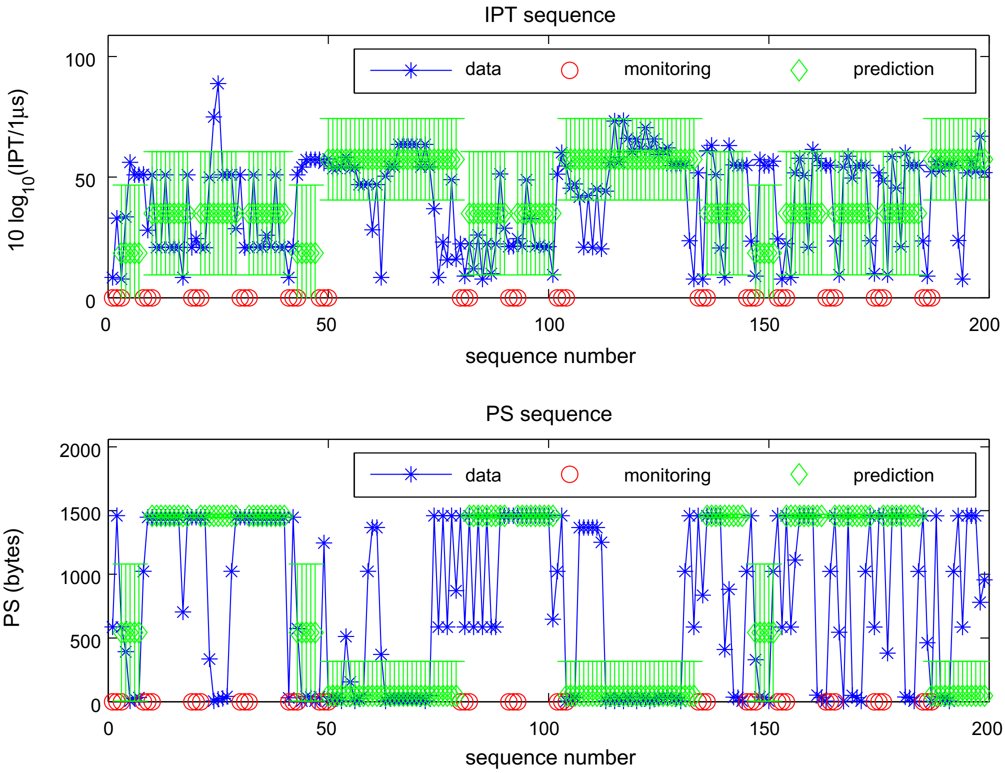


Fig. 8. Monitoring and prediction for SMTP: IPT (top) and PS (bottom).

eration of traffic patterns, and thus it may be integrated into a traffic generation or simulation framework.

5.1.2. HTTP traffic

In Fig. 5 the fitting of marginal distributions is shown. Note that IPT are spread over eight orders of magnitude, but the majority of them is concentrated approximately between 10 ms and 1 s. These values are compatible with RTTs found in Wide Area Networks. Indeed HTTP clients often perform a lot of subsequent requests to the same server. In the case of Web, for example, the first request of an HTML document is typically followed by more requests for the embedded objects. If such objects are small enough to be sent within one or few packets (as often is the case [46]), the requests are sent with intervals close to the RTT from the client to the server. As for the PS distribution, the HMM model captures the characteristics of real data but seems to slightly under-estimate small values in favor of larger ones.

Differently from the previous traffic typology HTTP does not present any dominant state while alternating various behaviors. It appears less regular also because states with all combination of small/large IPT and PS are present, as confirmed by the small cross-covariance. The correlation structures of HTTP, shown in Fig. 6, are very interesting. They present correlation at several lags with an oscillating behavior. The envelope decays faster for cross- than auto-covariances, and it is accurately captured by the trained model. It is worth saying that the model trained with single sessions (here not reported) captured the oscillating behavior too, while for the whole traffic, where different kinds of sessions are considered, this is quite hard and was not possible. Fig. 7 shows the results of synthetically generated HTTP traffic patterns.

5.1.3. AoM traffic

$N = 4$ states showed to be sufficient to capture the behavior of the data. Fig. 5 shows how PS are usually smaller than 20 bytes and concentrated around few close values, while on the contrary the IPT distribution presents a bi-modal behavior, with the 2 modes separated by more than 2–3 orders of magnitude. We found similar behaviors in other real-time strategy games, where stations typically send periodic update packets plus additional update packets when a user action must be immediately transmitted [47]. This evident link between user actions and packet-level traffic is probably one of the causes of the more randomness that was found in AoM traffic, when compared to the other sources considered. Table 8 shows more specifically how both the IPT modes are associated to the same range of PS, although the mode with lower IPTs (s_2) is more spread in terms of PS with respect to the mode with larger IPTs (s_1). In addiction, looking at steady-state probabilities and conditional durations in Table 8 we can affirm that s_1 and s_2 capture the 2 typical situations of real-time strategy games (periodic updates and user actions) while s_3 and s_4 are transient model introduced by the model. Fig. 5 confirms the previous analysis.

Fig. 6 shows the results obtained for auto- and cross-covariance. We found that all three covariances rapidly decay, an aspect that is well captured by the trained model. Also note (see cross-covariance at Lag 0) the presence of a small dependence between IPT and PS of the single packet, well captured by the trained model. This denotes the dominance of the large IPT–small PS mode.

Fig. 7 shows how the model is able to accurately reproduce the AoM traffic pattern.

5.1.4. MSN traffic

MSN presents some similar characteristics to SMTP: two main modes for both IPT and PS as shown in Fig. 5, captured by states s_1 and s_4 as shown in Table 9. The former accounts for low bitrate behavior, large IPT ($\sim$60 dBμ) and small PS ($\sim$0.1 KB), while the latter for high bitrate behavior, small IPT ($\sim$40 dBμ) and large PS ($\sim$1.3 KB). Again the negative cross-covariance in Fig. 6 confirms this kind of coupling between IPT and PS. Whereas auto-covariance reveals the presence of memory for IPT and PS characteristics. Fig. 6 shows an exponentially decaying trend for the data that is well-captured by the model, denoting the presence of a significant dependence between IPT–PS pairs of successive packets.

In Fig. 7, it can be seen that the model is able to accurately reproduce the MSN Messenger traffic patterns, replicating the two main IPT and PS modes. Also in the case of this traffic category, these results look promising as regards the model suitability for synthetic traffic generation [7] and simulation.

5.2. Prediction: some preliminary result

After the presentation of the performance of the HMM-based proposed model, here we show some preliminary result of its prediction capability. Indeed, the correlation structure of the various traffic typologies suggests to use the trained model for prediction purposes on a sample trace. The main objective is to show the capability of the model to provide the expected short-term future behavior of the traffic with sufficient accuracy. Such a characteristic results particularly appealing when thought as part of a more complex network-sensing and adaptive-management system. In order to give an idea of what kind of information the proposed approach could provide to higher-level applications, we performed (off-line) the following basic steps on the traces previously described:

- *Monitoring* – *W* samples (in terms of IPT–PS pairs) are observed iteratively to obtain an estimate of the current state via the Viterbi algorithm [25];
- *Prediction* – on the basis of the current state estimate and of the trained model parameters, the traffic is assumed to remain in that state (thus keeping conditional mean values for IPT and PS) for number of samples proportional to the conditional duration (Eq. (4)).

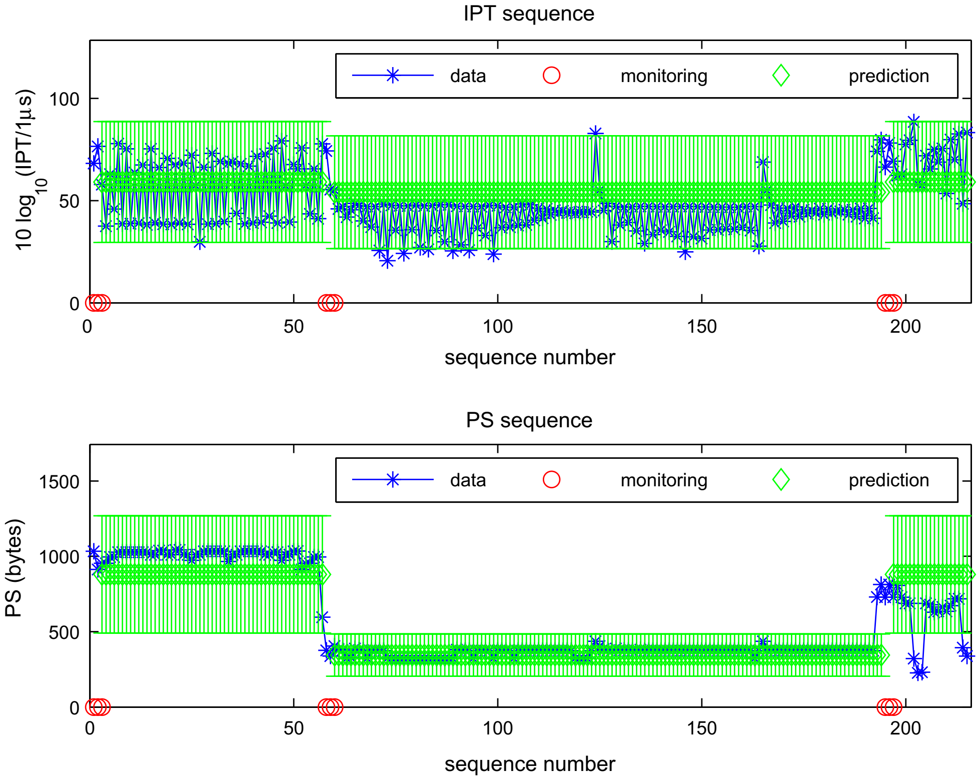


Fig. 9. Monitoring and prediction for HTTP: IPT (top) and PS (bottom).

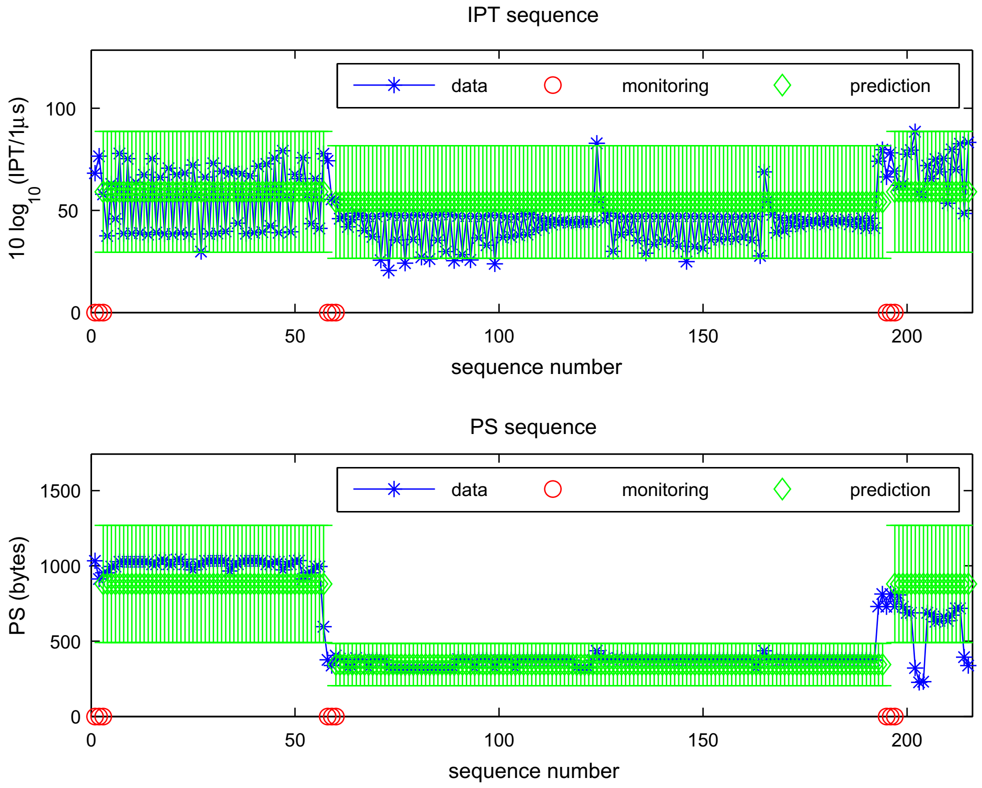


Fig. 10. Monitoring and prediction for AoM: IPT (top) and PS (bottom).

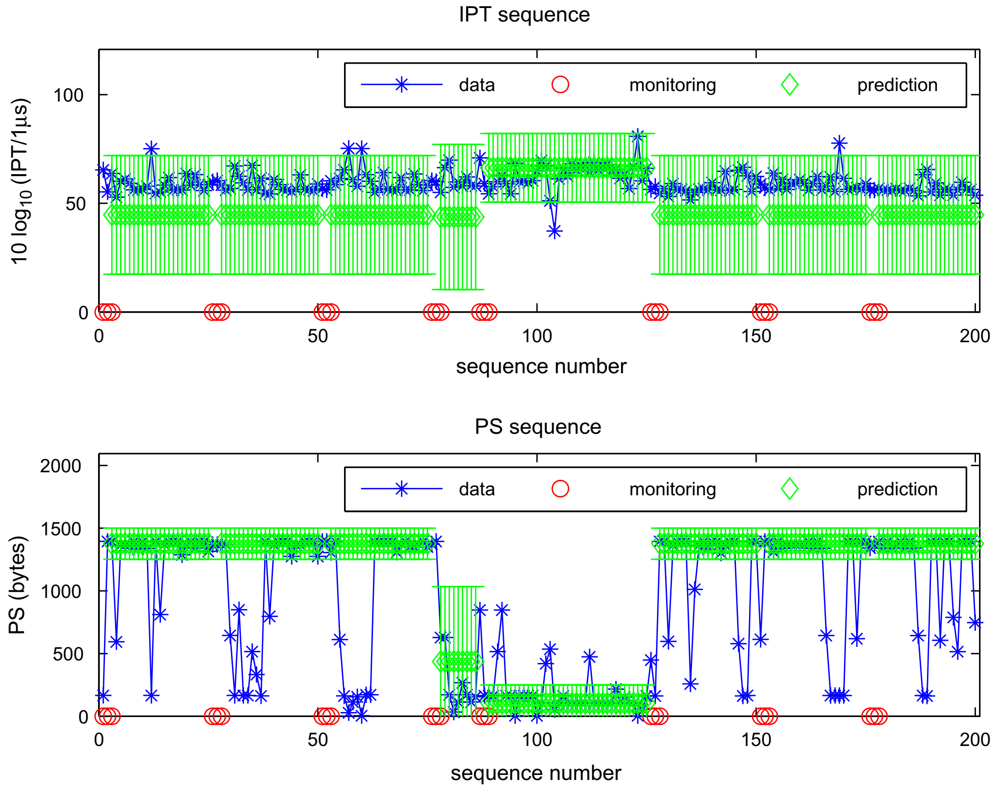


Fig. 11. Monitoring and prediction for MSN: IPT (top) and PS (bottom).

Figs. 10, 8, 9, 11 show results in the case of AoM, SMTP, HTTP and MSN traffic. We considered a $W = 3$-sample observation to obtain current-state estimate. Also, we assume that the traffic holds on conditional mean values, refer to Eq. (2), for a number of samples proportional to the conditional duration, refer to Eq. (4), of the state. The values of the conditional statistics, used for shown results, are those reported in Tables 6–9. Asterisks, circles, and diamonds represent data, monitored samples, and predicted samples, respectively. Also, for better precision, a confidence interval proportional to the conditional standard deviation, refer to Eq. (2), is reported with a green segment.

Comparing the frequent superposition between asterisks and diamonds, it can be noticed how the model captures and predicts the traffic dynamics for all the considered traffic typologies. Such a result is quite interesting, especially when looking at the source behavior when being in the states with large duration, i.e. s_1 for SMTP (whose conditional mean values are 57 dBμ and 48 bytes), s_2 for HTTP (whose conditional mean values are 54 dBμ and 345 bytes), s_1 for AoM (whose conditional mean values are 49 dBμ and 12 bytes), s_1 for MSN (whose conditional mean values are 66 dBμ and 104 bytes).

Also, the very small ratio between diamonds and circles gives an idea of the small amount of sensing that is needed to infer quite reliable information of what we should expect in the short term future behavior of the traffic. Note again that, due to the joint modeling we proposed, estimation of the state variable allows to infer knowledge about both IPT and PS expected behavior simultaneously.

Table 10 shows the relative mean square error (RMSE) and the percentage of monitoring data (MP) with respect the three different sizes of the monitoring window. It is apparent how, for all the considered traffic typologies, the RMSE is not significantly affected by the window size, both for IPT and PS, while obviously the MP is increasing with it. IPT prediction is very welle performing for all the considered traffic typologies, especially for AoM and MSN, while PS prediction is not very accurate for SMTP and MSN traffic. Indeed, the trained models for such applications present some discrepancy with the training data in PS pdf as well PS auto-covariance.

It is worth noticing that the experiment we have performed is just a simple not-optimum example of using the model. Another possibility could be building from the state estimated via the Viterbi algorithm an m-best list of most likely n-transitions. The real value of the model when used jointly with a monitoring algorithm, is the probabilistic representation in terms of the state matrix **A** of the possible evolution of the traffic.

Table 10
RMSE and MP for prediction with different sizes for the monitoring window

Traffic	Window size	IPT RMSE	PS RMSE	MP (%)
SMTP	$W = 3$	0.12	0.44	37
	$W = 5$	0.16	0.58	48
	$W = 7$	0.15	0.52	56
HTTP	$W = 3$	0.13	0.30	24
	$W = 5$	0.12	0.27	33
	$W = 7$	0.12	0.27	40
AoM	$W = 3$	0.027	0.13	27
	$W = 5$	0.027	0.13	38
	$W = 7$	0.027	0.14	46
MSN	$W = 3$	0.024	0.43	30
	$W = 5$	0.020	0.42	41
	$W = 7$	0.024	0.41	50

6. Discussion and conclusion

In this work, we proposed a HMM-based model of traffic sources at packet level. It jointly models IPT and PS of Internet applications traffic. It has been shown how the proposed HMM approach is able to capture the behavior of marginal distributions, mutual dependencies, and temporal structures of the traffic generated by a heterogeneous set of sources. The capability to accurately replicate and predict traffic makes the proposed approach quite promising.

Results obtained from four kinds of traffic sources, related to totally different Internet applications, have been analyzed. Empirical data clearly show that this heterogeneity is also reflected by the traffic that they generate at packet level. Indeed they differ for the behavior of the marginal distributions of IPT and PS but also for their correlation structure. As for the last point, it is worth noting that we found larger autocorrelations with a slower decay for SMTP, HTTP, and MSN traffic when compared to AoM. Such behavior can be partially explained by the influence of TCP end-to-end flow control, which introduces dependencies between IPTs. Indeed, while SMTP, HTTP, and MSN run over TCP, AoM traffic is carried by UDP packets. Furthermore, rigid application-level protocol rules of SMTP and HTTP induce more structure into their traffic patterns. On the other side, as regards AoM, the interaction of the gaming user introduces more randomness into the traffic. Again, we underline that the paper aims at modeling the average behavior of a single session. The study of the superposition of several sessions, generated by multiple sources may indeed lead to the generation of an aggregate traffic showing long range dependence and self similarity characteristics, but such investigation falls beyond the scope of the present work.

In all the cases the level of computational and structural complexity associated to the model is quite low. Training models for SMTP, HTTP, AoM and MSN required few iterations, and though SMTP and HTTP traffic present a much more complex structure, they only required one more state (with respect to AoM and MSN) for effective modeling. Then, the flexibility of an HMM approach, even when applied to a low-level traffic modeling, appears quite encouraging.

Concluding, it is worth highlighting that the more exciting result of the proposed model is, in our opinion, the capability to fit at the same time both IPT and PS statistics and dynamics, even if not obtaining extreme accuracy, of four different traffic sources with a relative small set of parameters. Benefits and possible applications of such modeling approach include: (i) a better understanding of source traffic dynamics (taking into account also temporal structures) related to different Internet applications; (ii) exploitation of the short-term prediction capabilities of the model; (iii) usage in traffic simulation and generation frameworks [7]; (iv) application for traffic classification purposes. Finally, we foresee the integration of the presented model within a larger analytical framework, based on HMMs, which includes modeling of heterogeneous packet channels [18,19].

References

[1] A. Dainotti, A. Pescapé, P. Salvo Rossi, G. Iannello, F. Palmieri, G. Ventre, An HMM approach to internet traffic modeling, in: Proc. of IEEE GLOBECOM, November 2006, pp. 1–6.

[2] A. Dainotti, A. Pescapé, G. Ventre, A packet-level characterization of network traffic, in: Proc. of IEEE CAMAD, June 2006, pp. 38–45.

[3] B.A. Mah, An empirical model of HTTP network traffic, Proc. of IEEE INFOCOM, vol. 2, April 1997, pp. 592–600.

[4] J. Cao, W.S. Cleveland, Y. Gao, K. Jeffay, F.D. Smith, M.C. Weigle, Stochastic models for generating synthetic http source traffic, in: Proc. of IEEE INFOCOM, vol. 3, 2004, pp. 1546–1557.

[5] R. Ohri, E. Chlebus, Measurement based e-mail traffic characterization, in: Proc. of Int. Symp. on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Philadelphia, PA, July 2005.

[6] W. Feng, F. Chang, W. Feng, J. Walpole, A traffic characterization of popular on-line games, IEEE/ACM Transactions on Networking 13 (3) (2005) 488–500.

[7] <http://www.grid.unina.it/software/ITG>, September 2007.

[8] J. Farber, Network Game Traffic Modelling, NetGames2002, Braunschweig, Germany, 2002. pp. 53–57.

[9] R. Bangun, E. Dutkiewicz, Modelling multi-player games traffic, in: Proc. of International Conference on Information Technology: Coding and Computing, 27–29 March 2000, pp. 228–233.

[10] M. Borella, Source models of network game traffic, Computer Communications 23 (4) (2000) 403–410.

[11] T. Lang, G.J. Armitage, P. Branch, H. Choo, A synthetic traffic model for half-life, in: Proc. of Australian Telecommunication Networks and Application Conference 2003 (ATNAC 2003), Melbourne, Australia, December 2003.

[12] P. Salvador, A. Pacheco, R. Valadas, Modeling IP traffic: joint characterization of packet arrivals and packet sizes using BMAPs, Elsevier Computer Networks 44 (October) (2004) 335–352.

[13] A. Klemm, C. Lindemann, M. Lohmann, Modeling IP traffic using the batch Markovian arrival process, Performance Evaluation Journal 54 (2) (2003) 149–173.

[14] J. Gao, I. Rubin, Multifractal analysis and modeling of long-range-dependent traffic, in: Proc. of IEEE ICC, June 1999, pp. 382–386.

[15] A.T. Andersen, B.F. Nielsen, A Markovian approach for modeling packet traffic with long-range dependence, IEEE Journal on Selected Areas in Communications 16 (5) (1998) 719–732.

[16] K. Salamatian, S. Vaton, Hidden Markov Modeling for network communication channels, in: Proc. of ACM SIGMETRICS 2001, vol. 29, 2001, pp. 92–101.

[17] W. Wei, B. Wang, D. Towsley, Continuous-time Hidden Markov Models for network performance evaluation, Performance Evaluation 49 (1–4) (2002) 129–146.

[18] P. Salvo Rossi, G. Romano, F. Palmieri, G. Iannello, Joint end-to-end loss-delay Hidden Markov Model for periodic UDP traffic over the Internet, IEEE Transactions on Signal Processing 54 (2) (2006) 530–541.

[19] G. Iannello, F. Palmieri, A. Pescapè, P. Salvo Rossi, End-to-end packet-channel Bayesian model applied to heterogeneous wireless networks, in: Proc. of IEEE GLOBECOM, November 2005, pp. 484–489.

[20] L. Muscariello, M. Mellia, M. Meo, M.A. Marsan, R. Lo Cigno, Markov models of internet traffic and a new hierarchical MMPP model, Computer Communications Journal 28 (16) (2005) 1835–1851.

[21] O. Rose, Simple and efficient models for variable bit rate MPEG video traffic, Performance Evaluation Journal 30 (1) (1997) 69–85.

[22] E. Costamagna, L. Favalli, F. Tarantola, Modeling and analysis of aggregate and single stream internet traffic, in: Proc. of IEEE GLOBECOM, December 2003, pp. 3830–3834.

[23] C. Wright, F. Monrose, G. Masson, HMM profiles for network traffic classification, in: Proc. of VizSEC/DMSEC, October 2004, pp. 9–15.

[24] <http://www.grid.unina.it/Traffic/>, September 2007.

[25] L.R. Rabiner, A tutorial on Hidden Markov Models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–285.

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

[27] L.A. Liporace, Maximum likelihood estimation for multivariate observations of Markov sources, IEEE Transactions on Information Theory IT-28 (5) (1982) 729–734.

[28] B.H. Juang, S.E. Levinson, M.M. Sondhi, Maximum likelihood estimation for multivariate mixture observations of Markov chains, IEEE Transactions on Information Theory IT-32 (2) (1986) 307–309.

[29] E. Çinlar, Introduction to Stochastic Processes, Prentice Hall, 1975.

[30] <http://nile.wpi.edu/downloads>, September 2007.

[31] R.B. Jennings, E.M. Nahum, D.P. Olshefski, D. Saha, Zon-Yin Shae, C. Waters, A study of Internet instant messaging and chat protocols, IEEE Network 20 (4) (2006) 1621.

[32] X. Zhen, G. Lei, J. Tracey, Understanding instant messaging traffic characteristics, in: Proc. of the 27th International Conference on Distributed Computing Systems (IEEE ICDCS 2007), Toronto, Canada, 25–29 June 2007.

[33] S. McCreary, K. Claffy, Trends in wide area IP traffic patterns – a view from Ames Internet Exchange, in: Proc. of ITC Specialist Seminar of Measurement and Modeling of IP Traffic, September 2000, pp. 1–11.

[34] <http://www.microsoft.com/games/ageofmythology/>, September 2007.

[35] <http://www.comscore.com/>, September 2007.

[36] <http://join.msn.com/messenger/overview>, September 2007.

[37] <http://www.microsoft.com/technet/prodtechnol/isa/2000/maintain/isaimsec.mspx>, September 2007.

[38] <http://www.hypothetic.org/docs/msn/general/overview.php>, September 2007.

[39] M. Claypool, The effect of latency on user performance in real-time strategy games, Elsevier Computer Networks 49 (1) (2005) 52–70.

[40] A. Dainotti, A. Botta, A. Pescapé, G. Ventre, Searching for invariants in network games traffic, in: Proc. of ACM Co-Next 2006 Student Workshop, Lisboa, Portugal, December 2006.

[41] S. McCanne, V. Jacobson, The BSD packet filter: a new architecture for user level packet capture, in: Proc. of Winter 1993 USENIX, January 1993, pp. 259–269.

[42] T. Karagiannis, A. Broido, M. Faloutsos, K. Claffy, Transport layer identification of P2P traffic, in: Proc. of ACM SIGCOMM IMC, October 2004, pp. 121–134.

[43] <http://www.wide.ad.jp/wg/mawi/>, September 2007.

[44] R. Caceres, P. Danzig, S. Jamin, D. Mitzel, Characteristics of wide-area TCP/IP conversations, ACM SIGCOMM Computer Communication Review 21 (4) (1991) 101–112.

[45] P. Danzig, S. Jamin, R. Caceres, D. Mitzel, D. Estrin, An empirical workload model for driving wide-area TCP/IP network simulations, Journal of Internetworking: Research and Experience 3 (1) (1992) 1–26.

[46] F.D. Smith, F.H. Campos, K. Jeffay, D. Ott, What TCP/IP procotol headers can tell us about the Web, in: Proc. of ACM SIGMETRICS, June 2001, pp. 245–256.

[47] A. Dainotti, A. Pescapé, G. Ventre, A packet-level model of Starcraft traffic, in: Proc. of IEEE Hot-P2P, July 2005, pp. 33–42.

[48] S.P. Pederson, M.E. Johnson, Estimating model discrepancy, Technometrics 32 (3) (1990) 305–314.

Alberto Dainotti is Ph.D. student in Computer Engineering and Systems at the Computer Science Department of University of Napoli "Federico II", Italy, where he received the M.S. Laurea Degree in Computer Engineering in 2004. His research interests fall in the areas of network measurements, traffic analysis, and network security.

Antonio Pescapé is Assistant Professor at the Department of Computer Engineering and Systems of the University of Napoli Federico II. He received the M.S. Laurea Degree in Computer Engineering and the Ph.D. in Computer Engineering and Systems at University of Napoli Federico II. His research interests are in the networking field with focus on models and algorithms for Internet Traffic, Network Measurement and Management of heterogeneous IP networks, and Network Security. He has co-authored a large number of journal and conference publications. He is IEEE member and he has served and serves on several conference technical program committees (IEEE Globecom, IEEE ICC, IEEE WCNC, IEEE HPSR, etc.) and has served as Guest Editor of the Special Issue of Computer Networks on "Traffic classification and its applications to modern networks".

Pierluigi Salvo Rossi was born in Naples, Italy, on April 26, 1977. He received the "Laurea" degree in Telecommunications Engineering (summa cum laude) in January 2002 and the Ph.D. in Computer Science in January 2005, both from the University of Naples "Federico II", Naples, Italy. In 2002, he worked as a Research Engineer at the CIRASS (Interdepartmental Research Center for Signal Analysis and Synthesis), University of Naples "Federico II", Naples, Italy. In 2003, he worked as Research Engineer at the Department of Information Engineering, Second University of Naples, Aversa (CE), Italy. In 2004, he was Visiting Research Engineer at the CSPL (Communications and Signal Processing Laboratory), Electrical and Computer Engineering Department, Drexel University, Philadelphia, PA, US. In 2005, he worked as Postdoc Research Engineer at the ITeM (Multimedia Information and Telematic National Laboratory), CINI (Italian University Consortium for Computer Science and Engineering), Naples, Italy. In 2006, he worked as Postdoc Research Engineer at the CRdC–ICT (Regional Institute for Research on Information and Communication Technology), Benevento, Italy. He is currently a Postdoc Research Engineer at the Department of Electronics and Telecommunications, Norwegian University of Science and Technology, Trondheim, Norway. Since 2004, he is Adjunct Professor at the Second University of Naples, Aversa (CE) Italy. His research interests fall within the areas of speech processing, communication system modeling, wireless communications.

Francesco Palmieri received his Laurea in Ingegneria Elettronica cum laude from Università degli Studi di Napoli Federico II, Italy, in 1980. In 1983, he was awarded a Fulbright scholarship to conduct graduate studies at the University of Delaware, Newark, where he received a M.S. degree in applied sciences and a Ph.D. in electrical engineering in 1985 and 1987, respectively. In 1981, he served as a 2nd Lieutenant in the Italian Army in fullfillment of draft duties. In 1982 and 1983, he was with the ITT firms: FACE SUD Selettronica in Salerno (currently Alcatel), Italy, and Bell Telephone Manufacturing Company in Antwerpen, Belgium, as a designer of digital telephone systems. He was appointed Assistant Professor in Electrical and Systems Engineering at the University of Connecticut, Storrs, in 1987, where he was awarded tenure and promotion to associate professor in 1993. In the same year, after a national competition, he was awarded the position of Professore Associato at the Dipartimento di Ingegneria Elettronica e delle Telecomunicazioni at Università degli Studi di Napoli Federico II, Italy, where he has been until October 2000. In February 2000, he was nominated Professore Ordinario di Telecomunicazioni after a national competition and appointed in November 2000 at Dipartimento di Ingegneria dell?Informazione, Seconda Università di Napoli, Aversa, Italy. His reaserch interests are in the areas of signal processing, communications, information theory and neural networks.

Giorgio Ventre is Professor of Computer Networks in the Department of Computer Engineering and Systems of the University of Napoli Federico II, where he is leader of the COMICS team. COMICS stands for Computers for Interaction and Communications and is a research initiative in the areas of networking and multimedia communications. After started ITEM, the first research laboratory of the Italian University Consortium for Informatics (CINI), He is now President and CEO of CRIAI, a research company active in the areas of Information Technologies. As leader of the networking research group at University of Napoli Federico II. He is principal investigator for several national and international research projects. His research interests are in the area of network protocols and architectures. He has co-authored more than 150 publications and he is member of the IEEE and of the ACM.