An MMPP Model of Internet Traffic

L. Muscariello, M. Mellia, M. Meo, M. Ajmone Marsan
Dipartimento di Elettronica, Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129 Torino, Italy

R. Lo Cigno
Dipartimento di Informatica e Telecomunicazioni, Università di Trento
Via Sommarive 14, 38050 Povo, Trento, Italy

Abstract—**In this paper, we propose an MMPP (Markov Modulated Poisson Process) traffic model that accurately approximates the LRD (Long Range Dependence) characteristics of Internet traffic traces over the relevant time scales.**

Using the notions of sessions and flows, the proposed MMPP model mimics the real hierarchical behavior of the packet generation process by Internet users.

Thanks to its hierarchical structure, the proposed model is both simple and intuitive: it allows the generation of traffic with desired characteristics by easily setting few input parameters which have an intuitive physical meaning.

Results prove that the queuing behavior of the traffic generated by the MMPP model is coherent with the one produced by real traces collected at our institution edge router under several different traffic loads.

Due to its characteristics, the proposed MMPP traffic model can be used as a simple and manageable tool for IP network dimensioning, design and planning.

I. Introduction

The fact that packet flows arriving at Internet routers (both edge and backbone) cannot be accurately modeled by Poisson processes is widely accepted, and has been discussed in a vast literature (see for example [1], [2], [3], [4], [5]). One of the main characteristics of Internet traffic, probably the one with the most impact on planning and dimensioning, is the Long Range Dependence (LRD) of the distribution of several traffic parameters (e.g., packet inter-arrival time, amount of data transferred per time unit, etc.). This LRD means that Internet traffic has some sort of *memory*; however, long term correlation properties, heavy tail distributions, and all other characteristics of Internet traffic, are meaningful only over a limited range of time scales. For instance, any correlation property on time scales smaller than the packet transmission time has no physical meaning. Similarly, heavy-tails of distributions describing file lengths, become meaningless beyond the limitations imposed by storage media. Although a number of traffic models have been derived by fitting real measured Internet traffic traces (we discuss some of them in Section II), they seldom allow the generation of traffic with desired characteristics. On the contrary, a model of Internet traffic, in order to be effectively used for network dimensioning, must be simple, easy to understand, and, most important, must be controlled through a small number of parameters, whose influence on the generated traffic is predictable, at least from a qualitative point of view.

This work was supported by the Italian Ministry for Education, University and Research through the TANGO project.

In this paper, we propose a simple Markovian model of Internet traffic, that matches very well the characteristic of the Internet traffic observed at an Edge Router, and resulting from the aggregation of many individual packet flows. The model is based on *Markov Modulated Poisson Processes* (MMPPs), and aims at providing a description of the traffic generation process as close as possible to the typical behavior of real Internet applications. The model employs the notion of three entities: *sessions*, *flows* and *packets*; which act at different time scales, and mimic the real behavior behind the interaction between users, protocols, and the network. The model has five parameters which are very simple to tune, so as to either fit the characteristics of a real trace, or generate traffic with known statistical properties. Three of the parameters are mapped directly on average traffic characteristics, such as the average load and the average flow size; two parameters are used to shape the correlation properties of the traffic. This tunability feature, combined with the simplicity of the model, makes it a very effective synthetic generator of aggregate Internet traffic, that can be used to predict the network behavior as the traffic characteristics change.

Finally, besides being very efficient to simulate in complex networking scenarios, the MMPP traffic model also allows analytical solutions of its queuing behavior to be derived in simple cases. Under some simple scenarios of practical interest, these analytical solutions are very efficient, and provide a useful alternative to simulations.

Summarizing, the contributions of this paper are the following:

- an MMPP model of Internet traffic is proposed
- the accuracy of the MMPP model is proved by comparing the characteristics of synthetic traffic to those of real traces
- the MMPP model is shown to be useful for network planning and dimensioning by comparing the performances of queues fed by either synthetic traffic or real traces
- an analytical model of a queue fed by the MMPP traffic model is solved

The rest of this paper is organized as follow. Section II shortly discusses the literature that influenced our research. Section III describes the measured data we used for traffic analysis and for the model validation. Section IV presents the MMPP model and its ties to measurable traffic characteristics. Section V discusses the performance of the model, highlighting its strengths, but also discussing its limitations. Section VI finally concludes the paper.

II. Related Work

In the early 90's, two seminal papers [1], [2] showed that traffic traces captured on both LANs and WANs exhibit long range dependence (LRD) properties, and self-similar characteristics at different time scales. Those discoveries spurred a significant research effort to understand data traffic in packet networks in general, and in the Internet in particular. In addition, the evidence of LRD and of self-similar properties in packet traffic drove many researchers to abandon the usual Markovian assumptions in favor of newer and more complex traffic models. A number of attempts were made to develop models for LRD data traffic. Here we briefly summarize some of the main approaches proposed in the literature.

Looking at packet traffic as a superposition of source-destination traffic flows, simple *ON/OFF models* (or *packet trains models*) were proposed as a first way to mimic LRD properties [3], [4]. Indeed, if the *ON* (*OFF*, or both) periods are generated according to heavy-tailed distributions, and the number of multiplexed flows is large, then the resulting aggregate traffic exhibits asymptotic self-similar properties with LRD behavior, as proved in [4].

An $M/G/\infty$ queue with infinite variance service time exhibits LRD properties in the number of active servers [6]. Since the heavy-tailed distribution of file sizes was measured on storage devices [5], [7], [8], and interpreting it as either the ON period duration or the service time of a file transfer, these two models have often been considered as a good explanation of LRD at packet level. At the same time, other recent studies [9], [10] indicate instead that traffic properties are rooted in the TCP congestion control mechanism, which induces LRD properties in the aggregate traffic resulting from the superposition of independent sources. Others underlined that TCP induces correlation at packet level on a limited range of time scales [11]. The statistical analysis of real traffic traces, due to the significant amount of collected data and of research projects, gave new impulse to traffic modeling. Among the numerous generic LRD models proposed in the literature, *Fractional Brownian Motion* (FBM) received a lot of attention, since its Gaussian nature helps in the study of the queuing behavior [12], [13]. However, this model presents a restrictive correlation structure, that fails to capture the short-term correlation of real traffic and its rich scaling behavior. Therefore, many research efforts were devoted to *Multifractal* models (see [14]), whose attractiveness is due to their rich scale-invariance properties. Indeed, previous analytical works, such as [7], [8], [15], [16], [17] suggested Multifractal models as possibly being the best fit to measured data. 'Cascades', a multifractal subclass, [18], [19] are also extensions of self similar models and capture traffic behavior at all meaningful time scales. While these models give good approximations of the LRD properties of Internet traffic, they are difficult to manage, due to their analytical complexity.

Wavelet decomposition has been widely used as a natural approach to study scale invariance, but only recently introduced in the field of data networks. There are many examples of measurement based traffic models, which try to fit the LRD properties of real traffic (see for example [20], [21]). While these models are computationally very efficient, they are complex and difficult to tune, due to the lack of a mapping between the traffic parameters and the model coefficients.

Chaotic map models [22], [23] were proposed as a deterministic evolution of systems governed by a set of behavioral rules. The derived models are simple, but it is often difficult to understand the relationship between the model and real traffic parameters.

FARIMA models [24] are widely used in video trace modeling, and can be used to generate LRD sequences. These models are derived by filtering white Gaussian noise, and capture both the short and the long period correlations of traffic. However, the models are quite complex, and their structure makes it very hard to understand the relationship among the filter coefficients and the real traffic data.

All these traffic characterization works deviate considerably from classical Markovian models which continue to be widely used for performance evaluation purposes with good results [25], [26], [27], [28]; in the above works for example, the Markov Modulated Poisson Process (MMPP), has been considered as the best Markov process to emulate LRD [26] and scale invariance [25], multifractality in particular. However, in [27], [28] it was also pointed out that, obviously, an MMPP does not exhibit long-term correlation; the authors therefore defined the *local* Hurst parameter, using an approximate LRD definition, valid on a limited range of time scales.

Another approach to model Internet traffic involves the emulation of the real hierarchical nature of network dynamic; in [29], each of the model components was fitted to real objects, such as the distribution of both TCP flows and web pages size, and the arrival distribution of pages and flows. In this paper, we follow a similar approach, but instead of trying to fit all possible distributions to the measured one, we use a much simpler Markovian definition.

In spite of the many proposed traffic models with LRD characteristics, very little work appeared in the field of network design and planning, or network performance analysis, based on LRD traffic models. This is mainly due to the difficulty in handling the complex mathematical structure of the stochastic processes on which those traffic models are based. Moreover, in [30] it was recently shown that the long-term correlation of traffic beyond a certain threshold does not influence the performance of a system, so that simple models where correlation is limited (such as MMPP models) can be successfully employed.

The results in [31], [32] also provide support to the possibility of using Markovian traffic models in packet networks, showing that bandwidth sharing in packet networks is insensitive to both the flow size, and the flow arrival process, under the quite commonly accepted assumption (see also [2], [3]) that session arrivals are Poisson.

All of these different approaches reach similar conclusions using different techniques. The objective always is to take into account as accurately as possible the real traffic behavior, in order to be able to: i) use more reasonable tools for network planning, and ii) explain the links between causes and effects of network traffic phenomena.

III. Traffic Measurement and Analysis

In order to collect traffic traces, we observed the data flow on the Internet access link[1] of our institution, i.e., we focus on the data flow between the edge router of Politecnico di Torino and the access router of GARR/B-TEN [33], the Italian and European Research network. For traces collection and processing we used `tcpdump` [34] and `Tstat` [35], [36]. `Tstat` is a new software tool developed at Politecnico di Torino, which analyzes traces, and derives traffic characteristics at both the IP and TCP levels. For the analysis at the TCP level, `Tstat` rebuilds each TCP connection status by looking at the TCP header in the forward and backward packet streams. In order to do so, `Tstat` requires as input a trace collected on an edge node, such that both data segments on the forward stream and ACKs on the backward stream can be analyzed. When `Tstat` observes a TCP connection opening and closing, it marks the flow as *complete*, and proceeds by analyzing it. Additional information about `Tstat` and statistical analysis performed on collected traces can be found in [35] and [36].

The Politecnico LAN comprises approximately 7,000 hosts; most of them act as clients, but several servers are also regularly accessed from outside hosts. Data were collected on files storing 6 or 3 hours long traces (to avoid exceeding the File System limitation on the file size), for a total of more than 100 Gbytes of compressed data. Traces were collected during different periods, which correspond to different phases of the network topology evolution. In this paper, we present results considering two periods which are characterized by a significant upgrade in network capacity:

- April 2000, from 4/11/2000 to 4/14/2000: the bandwidth of the access link was 4 Mbit/s, and the link between the GARR network and the corresponding US peering was 45 Mbit/s
- February 2001, from 2/1/2001 to 2/19/2001: the bandwidth of the access link was 16 Mbit/s, and of the link between the GARR network and the corresponding US peering was 622 Mbit/s.

The campus access link was a bottleneck during April 2000, while it was not during February 2001. The same consideration applies to the GARR-US peering capacity, which plays a key role, since most of the traffic comes from US research sites.

Among all the traces we collected, we report here results from four traces, which we consider representative of different network scenarios. Table I summarizes the key parameters of the selected traces; the last two columns report the number of samples in a trace, i.e., the number of IP packets and of TCP flows.

Since our campus network can be mainly considered as a "client" network, i.e., hosts in the network are mainly destinations of information, in the remaining of this paper we will present results considering incoming streams of data only, both at the TCP flow level and at the IP packet level.

A. Definitions

Several different definitions of LRD exist (which in general are not equivalent; see for example [37]). We recall in this section the definition we use in this paper, which is the one proposed in [38].

[1]The data-link level exploits an AAL-5 ATM virtual circuit (OC-3).

TABLE I
Summary of the analyzed traces

Name	Date	Start time	Stop time	IP packets (10^6)	TCP flows (10^3)
Peak'01	2 Feb 01	10:52	13:52	11	540
Night'01	2 Feb 01	04:52	07:52	0.43	30
Peak'00	13 Apr 00	08:10	14:10	12	564
Night'00	13 Apr 00	02:10	08:10	0.92	79

Definition (Long Range Dependence)
Let $\{X_k\}_{k\in\mathbb{Z}}$ be a wide sense stationary random sequence, μ its mean, $\gamma(n)$ its autocovariance function, and $f(\nu)\;\; \nu \in [-\pi,\pi]$ its spectral density,

$$\mathbb{E}[X_k] = \mu \tag{1}$$

$$\mathbb{E}[(X_k-\mu)(X_{k+n}-\mu)] = \gamma(n), \;\; n \in \mathbb{Z} \tag{2}$$

$$f(\nu) = \frac{1}{2\pi}\sum_{n=-\infty}^{+\infty} e^{-i\nu n}\gamma(n) \tag{3}$$

$$\gamma(n) = \int_{-\pi}^{\pi} f(\nu)e^{-i\nu n}d\nu \tag{4}$$

$\{X_k\}_{k\in\mathbb{Z}}$ is said to be *Long-Range Dependent* if

$$\gamma(n) \sim a n^{\alpha-1}, \quad n\to\infty, \;\; \alpha \in (0,1) \tag{5}$$

or if

$$f(\nu) \sim c|\nu|^{-\alpha}, \nu \to 0, \;\; \alpha \in (0,1) \tag{6}$$

where $f(x) \sim g(x)$ as $x \to x_0$ means $\lim_{x\to x_0} f(x)/g(x) = 1$. Equations (5) and (6) are equivalent if $\gamma(n)$ is monotone. In the following we use the definition in (6), that is based on two parameters: (α, c). The parameter $\alpha \in [0,1)$ is the dimensionless scaling exponent, and describes the "intensity" of LRD; for a non-LRD stationary process, $\alpha = 0$ at large scales. The parameter $c \in \mathbb{R}^+$ has the same dimension of the variance of the process and describes the quantitative aspect of LRD often referred to as the *LRD size*. LRD implies that the sum of correlations over all lags is infinite; however, individually, their sizes at large lags are proportional to c, and can be arbitrarily small. LRD is usually associated with *self similar processes with stationary increments* (H-sssi) defined as follows.

Definition (Self-Similarity)
Let $\{Y(t), t\in\mathbb{R}\}$ be a random process; $\{Y(t)$ is said to be *H-self-similar* (H-ss) if

$$\{Y(t), t\in\mathbb{R}\} \stackrel{d}{=} \{r^{-H}Y(rt), t\in\mathbb{R}\}, \forall r \in \mathbb{R}^+, \quad H>0$$

[2] If $Y(t)$ has stationary increments $\{X_k\}_{k\in\mathbb{Z}}$, $X_k = Y(k-h) - Y(k)$ is LRD with

$$\alpha = 2H - 1 \qquad \text{if } H \in (0.5, 1) \tag{7}$$

The parameter H is known as the *Hurst parameter*. When considering the process of the increments of a self similar process with stationary increments, relationship (7) holds; hence, it is common practice (though not completely proper) to use the parameter H when discussing LRD, and we stick to this practice. Clearly, a non-LRD process has $H = 0.5$, while Hurst parameters larger than 0.6–0.8 are normally assumed as an indication of strong LRD.

[2]The equality is for finite dimensional distributions

B. Trace Analysis

In order to estimate the LRD properties of the stochastic process of interest, we use the wavelet-based approach developed in [38], [39] and the the tools presented there, that are usually referred to as the *AV* estimator. We also employ the code made available from the authors in [40]. Other approaches can be pursued to analyze traffic traces, but the wavelet framework has emerged as one of the best estimators, as it offers a very versatile environment, as well as fast and efficient algorithms.

Since traffic traces are finite, and their asymptotic behavior cannot be derived, we always limit the analysis between two scales $(j_{\inf}, j_{\sup})$. In order to evaluate LRD parameters, we use the Log-Scale Diagram, which is essentially a log-log plot of the mean square values estimates of the wavelet coefficients x_n^j versus the scale j. Since 2^{-j} has the dimension of a frequency, j is generally called *octave*. Through the Log-Scale Diagram, it is possible to identify the presence of LRD and determine the cutoff scales $(j_{\inf}, j_{\sup})$ at which LRD 'begins' and 'ends.' Within these scales, an LRD process Log-Scale Diagram is linear with coefficient α. Indeed, for all processes, $j_{\sup}$ is limited by the trace length, and $j_{\inf}$ corresponds to a scale of a few hundreds milliseconds, in our measures.

Among all the possible metrics that can be derived from the traces, we selected as most representative of the traffic characteristics:

- the (packet and flow) inter-arrival time processes $I(k)$
- the (packet and flow) counting processes $N_T(n)$, obtained by counting the number of arrivals in a time interval $[nT, (n+1)T)$; we use three values of T: $1, 0.1, 0.01$ s.

Combining the three tools (`Tstat`, `tcpdump`, and *AV*) we analyzed the metrics defined above at both the TCP flow and IP packet levels.

C. Flow and Packet Level Analysis

Given a trace and a process under analysis, the *AV* estimator produces estimates of three main parameters: the Hurst parameter, the c factor and the mean value. Estimates are denoted by $\hat{H}_f$, $\hat{c}_f$, and $1/\hat{\Lambda}_f$ when traces are analyzed at the flow level; and by $\hat{H}_p$, $\hat{c}_p$, and $1/\hat{\Lambda}_p$ for the packet level analysis. Results for the four traces we consider are reported in Tables II and III for the flow level analysis and for the packet level analysis, respectively. In the tables, as previously mentioned, I denotes the interarrival time process, N_T the counting process in time intervals whose duration is equal to T.

Apart from the obvious consideration that during peak hours arrival rates are much higher than during nights, a few observations are in order. First of all, notice that all processes show similar values of the Hurst parameter, ranging from 0.71 to 0.88. Indeed, $\hat{H}_f$ is almost independent from the considered trace or process; while $\hat{H}_p$ is slightly higher during peak hours (around 0.88) than during nights. These results hint to the fact that LRD in packet networks is probably not due to high load. Moreover, since the network characteristics of the four selected traces are very different (different link speeds, different loads, different patterns between peak and night hours), this can be taken as a strong indication that LRD is an intrinsic characteristic of the Internet traffic and is not induced by network conditions; this is coherent with the "Pareto effet" due to file size distributions, assumed as a good explanation of LRD at packet level. A second consideration concerns c, whose value is extremely variable, and clearly connected to the absolute magnitude of the analyzed process (indeed, it is connected to the mean square value of the process itself). The last consideration is that the characteristics of the measured traces do not change significantly from '00 to '01; thus, in the sequel we will only present results for these latter traces, that are more recent.

TABLE II
FLOW LEVEL ANALYSIS OF TRACES

Trace	Peak'01			Night'01		
	$\hat{H}_f$	$\hat{c}_f$	$1/\hat{\Lambda}_f$	$\hat{H}_f$	$\hat{c}_f$	$1/\hat{\Lambda}_f$
$I\ [ms]$	0.74	82.4	20.01	0.86	8491	358.9
N_{1s}	0.76	59.4	49.9	0.76	1.66	2.79
N_{100ms}	0.75	2.01	4.99	0.73	0.07	0.28
N_{10ms}	0.74	0.07	0.49	0.80	0.001	0.028
Trace	Peak'00			Night'00		
	$\hat{H}_f$	$\hat{c}_f$	$1/\hat{\Lambda}_f$	$\hat{H}_f$	$\hat{c}_f$	$1/\hat{\Lambda}_f$
$I\ [ms]$	0.76	275.1	39.6	0.74	2414	271.6
N_{1s}	0.75	28.4	25.9	0.78	1.54	3.68
N_{100ms}	0.74	0.79	2.53	0.76	0.06	0.37
N_{10ms}	0.75	0.015	0.25	0.78	0.001	0.04

TABLE III
PACKET LEVEL ANALYSIS OF TRACES

Trace	Peak'01			Night'01		
	$\hat{H}_p$	$\hat{c}_p$	$1/\hat{\Lambda}_p$	$\hat{H}_p$	$\hat{c}_p$	$1/\hat{\Lambda}_p$
$I\ [ms]$	0.87	0.01	0.89	0.71	$5 \cdot 10^{-4}$	0.025
N_{1s}	0.88	5232	1113	0.73	457.8	40.06
N_{100ms}	0.88	91.4	111.3	0.72	16.6	4.00
N_{10ms}	0.88	1.50	11.1	0.76	0.30	0.4
Trace	Peak'00			Night'00		
	$\hat{H}_p$	$\hat{c}_p$	$1/\hat{\Lambda}_p$	$\hat{H}_p$	$\hat{c}_p$	$1/\hat{\Lambda}_p$
$I\ [ms]$	0.84	0.17	2.25	0.84	40.54	15.74
N_{1s}	0.86	504.19	444.75	0.83	133.93	63.51
N_{100ms}	0.87	14.50	44.49	0.83	3.61	6.35
N_{10ms}	0.88	0.23	4.45	0.87	0.04	0.63

For the model development, besides the estimates mentioned above, we also derived from the traces:

$\hat{N}_p$: the average number of packets per flow

$\hat{\lambda}_p$: the packet generation rate of active flows (obtained as the ratio between $\hat{N}_p$ and the average flow duration).

IV. THE MMPP TRAFFIC MODEL

Today, Internet traffic is mainly generated by data transfers that use the TCP protocol at the transport layer. We derive our model by keeping in mind that, in layered architectures, the human actions on a terminal interface cause a sequence of events and behaviors of the protocols at the various layers of the protocol stack. For example, a "click" on a web link causes the generation of a request at the application level (i.e., an HTTP request), which is translated into many transport level connections (TCP flows); each connection generates a sequence of data

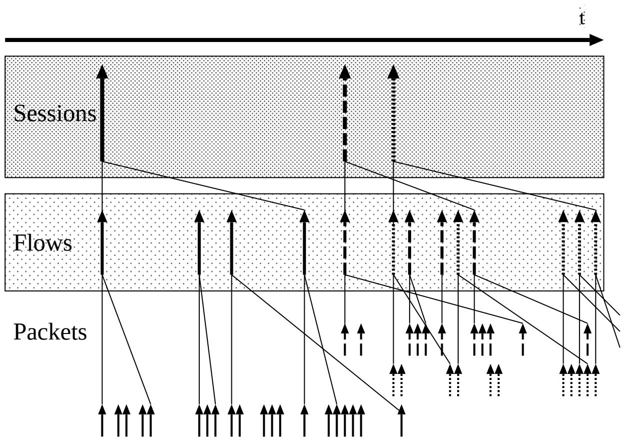

Fig. 1. Sessions, flows, and packets as seen by the model

segments that are transported by the network through IP packets.

According to this view, we propose a model whose behavior is driven by entities which act at three different time scales. At the application level, *sessions* correspond to bunches of information transfers. Sessions generate *flows* which correspond to TCP connections. Each flow generates a sequence of *packets* injected into the network. Figure 1 sketches a realization of the model; three sessions arrive, each one generates a given number of flows, and each flow, in its turn, generates a given number of packets, that will be multiplexed on links along the source-destination paths.

In order to derive a model which emulates the behavior of a given real trace, we now need to map the three model entities (sessions, flows and packets) into entities which can be observed in real traces. Packets and flows can be easily identified on the real trace; in particular, a flow is a single TCP connection, which starts by the three-way-handshake procedure and ends by the closing procedure[3]. On the contrary, it is more difficult to provide a specific and unique definition of a session. Indeed, many different definitions of a session could be proposed on the basis of traces. All the web pages downloaded by a user from the same web server in a limited period of time can form a session; a ftp connection from a user that requests many files from a server can form a session; all the e-mail messages generated by a user that replies to all the previously downloaded e-mails, or even the user activating its connection to the Internet (for example by switching on its computer), are all possible definitions of a session. Thus, we have the following problem. On the one hand, sessions are difficult to define, and to recognize in real traces. On the other hand, we need a notion of session in order to account for correlation over long time scales, which a model based on flows and packets alone cannot catch. We resort to defining a session as a generic set of correlated flows that are submitted to a network interface; then, we use the flow and packet levels of the model to fit the metrics which are easy to measure on real traces, and we specify the model session level so that the LRD of the real traces is accurately approximated.

[3] In case no packets are observed for more than 30 min, the flow is declared closed as well.

We now make the following assumptions concerning the behavior of sessions, flows and packets:

- Sessions are generated according to a Poisson process with rate λ_s; each session starts by generating a new flow and ends when it generates the last flow belonging to the session.
- The number of flows generated by a session is a geometrically distributed random variable with mean value equal to N_f.
- Flows belonging to the same session are generated according to a Poisson process with rate λ_f; each flow starts by generating a packet and ends after having generated the last packet of the flow.
- The number of packets generated by a flow is a geometrically distributed random variable with mean value equal to N_p.
- Packets belonging to the same flow are generated according to a Poisson process with rate λ_p.

Due to the above assumptions, both the packet arrival process and the flow arrival process are MMPP, whose memory is given by two variables: the number of active flows, which accounts for short term correlation, and the number of active sessions, which determines correlation over long time lags. Observe that the model analysis can also use the same formalism used in [41]

A. Setting the Model Parameters

The MMPP model is completely described by five parameters:

λ_s : the arrival rate of new sessions
λ_f : the flow arrival rate per active session
λ_p : the packet arrival rate per active flow
N_f : the average number of flows per session
N_p : the average number of packets per flow.

In order to tune the model so that the generated synthetic traffic emulates the characteristics of a real trace, we need to properly set the model parameters. The parameters N_p and λ_p are simply set so as to match the packet and flow behaviors of a given trace; i.e., they are set equal to the measured average number of packets per flow and average packet arrival rate per flow. For what concerns N_f and λ_s, since they are related to the session behavior, they are harder to measure from traces. Thus, we set N_f and λ_s so as to match the Hurst parameters $\hat{H}_f$ and $\hat{H}_p$; this is done by means of an iterative procedure described below. Finally, given the session behavior, the parameter λ_f, the flow arrival rate per session, is simply set to match the average flow arrival rate observed from the traces. The fitting procedure is summarised in Fig. 2 (notice that, for our convenience, in the fitting procedure we normalize the session arrival rate λ_s to the flow arrival rate per session λ_f, and denote the normalized arrival rate by C):

The selection of N_f and C by means of the fitting procedure at steps 4–7 can be performed according to different definitions of accuracy of the fit and to different criteria for the assignment of new values to the parameters. The detailed procedure which we followed is reported in Fig. 3.

The criteria to assign new values to N_f and C were chosen after having studied the sensitivity of the Hurst parameters H_f and H_p to changes of N_f and C by means of the graphs shown in Fig. 4. The Hurst parameter at both the flow and packet levels

1. From the traffic traces estimate $\hat{H}_p$, $\hat{H}_f$, $\hat{\Lambda}_f$, $\hat{\lambda}_p$, $\hat{N}_p$
2. Set $N_p = \hat{N}_p$ and $\lambda_p = \hat{\lambda}_p$
3. Set the initial values $N_f = 1$ and $C = 1$ (C is defined as $C = \lambda_s/\lambda_f$)
4. Compute $\lambda_s = \frac{\hat{\Lambda}_f}{N_f}$ and $\lambda_f = \frac{\lambda_s}{C}$
5. Generate a synthetic sequence with the same number of samples as the real trace
6. Estimate the Hurst parameter at both packet and flow level of the synthetic trace and compare them with $\hat{H}_p$, $\hat{H}_f$
7. If the fitting is good, the procedure ends else assign new values to N_f and C and go to 4

Fig. 2. Fitting procedure to derive the MMPP model parameters from a measured trace

$C = 1$; $N_f = 1$;
$eps_f = eps_p = 0.05$; fit = 0;
while !(fit){
  generate a synthetic sequence and estimate H_f, H_p;
  fit = ($|H_f - \hat{H}_f| < eps_f$) &&($|H_p - \hat{H}_p| < eps_p$);
  if ($H_f < \hat{H}_f$) then $N_f = N_f + 5$;
  else if ($H_f > \hat{H}_f$) then $N_f = N_f - 1$;
  if ($H_p > \hat{H}_p$) then $C = 3 * C$;
  else if ($H_p < \hat{H}_p$) then $C = C/2$;
}

Fig. 3. Selection of new parameters in the iteration

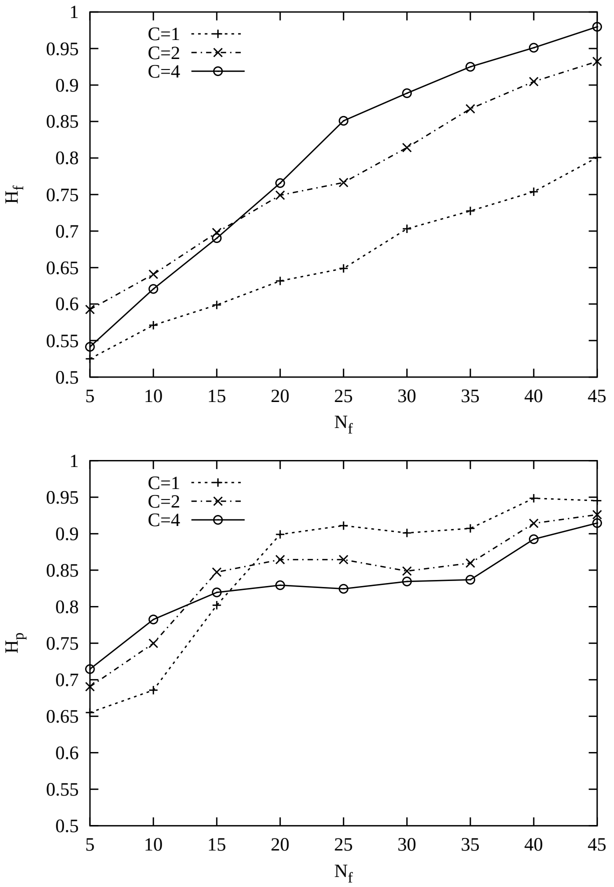

Fig. 4. Impact of N_f on the Hurst parameters of the flow arrival process, for different values of the normalized session arrival rate $C = \lambda_s/\lambda_f$

increases as N_f increases, consistently with the intuition that a larger value of N_f introduces a higher degree of memory in the system. Moreover, at the packet level, there is a higher degree of memory and correlation since packets are generated by flows which are generated by sessions. Let us now focus on C. The larger C is, the more bursty the generation of flows per session is. The influence of C on the Hurst parameter is quite complex. At the flow level, a higher degree of burstiness tends to induce a larger value of the Hurst parameter, while the opposite is true at the packet level.

The fitting procedure requires only a few iterations (approximately 10 in our tests). In order to verify that the synthetic traffic generated by the model accurately emulates the real traces, we report in Table IV the characteristics of the synthetic traces measured by the *AV* tool when the model parameters are fitted to the '01 traces; the values must be compared with those of Tables II. Observe that, thanks to the fitting procedure, the Hurst parameters are well matched, while the values for the c parameters are less accurate, though the qualitative behavior is the same.

B. The Modulating Markov Chain

The Continuous-Time Markov Chain (CTMC) which modulates the packet and flow arrival processes is defined by the state variable $\overline{s} = (n_f, n_s)$, where n_f and n_s denote the number of active flows and the number of active sessions, respectively. A state-transition diagram of the CTMC model of the modulating process is reported in Fig. 5.

The transition from state (n_f, n_s) to state $(n_f - 1, n_s)$ corresponds to the termination of a flow; its rate is $n_f\mu_f$, where $\mu_f = \lambda_p/(N_p - 1)$. The generation of a new flow makes the chain move from state (n_f, n_s) to state $(n_f + 1, n_s)$ with rate $\beta n_s \lambda_f$ if the flow is not the last one of the session, and to state $(n_f + 1, n_s - 1)$ with rate $(1 - \beta)n_s\lambda_f$ if the flow is the last one. The probability β that a generated flow is the last one of a session is given by $\beta = 1 - 1/N_f$. In state (n_f, n_s) a session starts with rate λ_s and generates a new flow. If the session is composed of one flow only, the chain moves from (n_f, n_s) to $(n_f + 1, n_s)$ with rate $(1 - \beta)\lambda_s$; otherwise, the chain moves from (n_f, n_s) to $(n_f + 1, n_s + 1)$ with rate $\beta\lambda_s$.

The infinitesimal generator of the CTMC is infinite due to the unbounded values that n_s and n_f can take. Thus, in order to analyze the property of the MMPP or to evaluate the performance of a queue fed by the synthetic traffic generated by the MMPP we have two alternatives: i) we resort to simulation, ii) we truncate the CTMC so that the infinitesimal generator matrix becomes finite. In what follows we consider both cases. The criterion used for truncating the CTMC is described in Section IV-C. For the moment, assume that the CTMC has been truncated so that the number of active flows varies in the range $[f_m, f_M]$ and the number of active sessions varies in $[s_m, s_M]$.

We denote by $\{J(t), t \in \mathbb{R}^+\}$ the finite CTMC ob-

TABLE IV
MODEL: FLOW AND PACKET LEVEL RESULTS FITTING THE '01 TRACES

Trace	Peak'01			Night'01		
	H_f	c_f	$1/\Lambda_f$	H_f	c_f	$1/\Lambda_f$
$I\ [ms]$	0.74	19.2	20.1	0.84	7799	356.8
N_{1s}	0.71	76.7	49.8	0.82	0.51	2.81
N_{100ms}	0.71	2.27	4.98	0.83	$7.6 \cdot 10^{-3}$	0.28
N_{10ms}	0.78	0.013	0.50	0.79	$1 \cdot 10^{-4}$	0.03
Trace	Peak'01			Night'01		
	H_p	c_p	$1/\Lambda_p$	H_p	c_p	$1/\Lambda_p$
$I\ [ms]$	0.84	0.038	0.90	0.82	$1 \cdot 10^{-4}$	0.025
N_{1s}	0.87	5943	1113	0.87	35.2	40.06
N_{100ms}	0.82	178.2	111.4	0.79	2.05	4.01
N_{10ms}	0.84	3.13	11.14	0.86	$5 \cdot 10^{-3}$	0.41

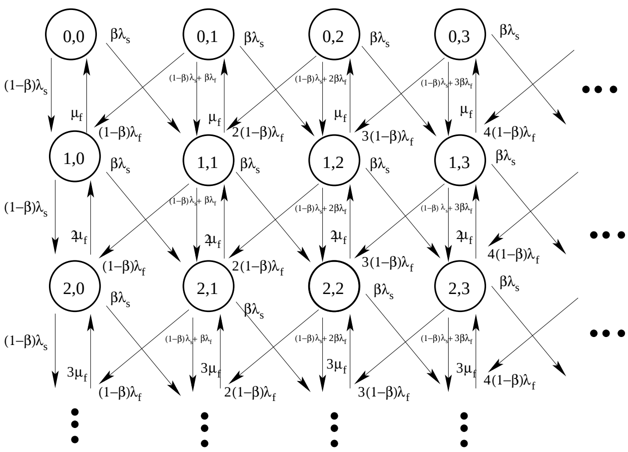

Fig. 5. State-transition diagram of the Continuous-Time Markov Chain that modulates arrivals

tained by truncating the MMPP according to the above ranges. The state space is given by $S = \{(n_f, n_s) \in \mathbb{N}^2, f_m \le n_f \le f_M, s_m \le n_s \le s_M\}$. The infinitesimal generator $Q \in \mathbb{R}^{n \times n}$ with $n = (f_M - f_m + 1) \cdot (s_M - s_m + 1)$ is given by,

$$Q = \begin{pmatrix} Q_{f_m} & Q^+ & 0 & \cdots & 0 \\ (f_m+1)Q^- & Q_{f_m+1} & Q^+ & \cdots & \vdots \\ 0 & (f_m+2)Q^- & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \cdots & \ddots & Q_{f_M-1} & Q^+ \\ 0 & \cdots & \cdots & f_M Q^- & Q_{f_M} \end{pmatrix}$$

$Q^- = \mu_f\, I_s$,
$Q^+ = (1-\beta)\lambda_f\, N_s^- + \beta\lambda_f\, N_s + (1-\beta)\lambda_s\, I_s + \beta\lambda_s\, I_s^+$

$$I_s^+ = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & 0 & 1 & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & 0 \end{pmatrix}$$

$$N_s^- = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ s_m+1 & 0 & \cdots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & s_M & 0 \end{pmatrix}$$

$I_s, I_s^+, N_s, N_s^- \in \mathbb{R}^{(s_M - s_m + 1) \times (s_M - s_m + 1)}$, I_s is the identity matrix. $I_s^- = I_s^{+T}$
$N_s = diags\{s_m, s_m+1, \cdots, s_M\}$[4],
$Q_i = diags\{-(s_m\beta\lambda_f + \lambda_s + i\,\mu_f), -[(s_m+1)\lambda_f + \lambda_s + i\,\mu_f], \cdots, -[(s_M-1)\lambda_f + \lambda_s + i\,\mu_f], -[s_M\lambda_f + (1-\beta)\lambda_s + i\,\mu_f]\}$, for $f_m < i < f_M$
$Q_{f_m} = diags\{-(s_m\beta\lambda_f + \lambda_s), -[(s_m+1)\lambda_f + \lambda_s], \cdots, -[(s_M-1)\lambda_f + \lambda_s], -(s_M\lambda_f + \lambda_s)\}$
$Q_{f_M} = diags\{-f_M\mu_f, -f_M\mu_f, \cdots, -f_M\mu_f\}$. Q is an homogeneous irreducible infinitesimal generator if $\beta \ne 0, 1$. The steady-state probability vector π is given by

$$\pi Q = 0, \quad \pi\, e = 1$$

where $e = (1, 1, \cdots, 1)^T$.

The rate matrix is $\Lambda \in \mathbb{R}^{n \times n}$, with $\Lambda = \lambda_p\, diags\{f_m\, I_s, (f_m+1)\, I_s, (f_m+2)\, I_s, \cdots, f_M\, I_s\}$.

C. Truncating the Modulating CTMC

To truncate the modulating CTMC, we have to trade-off between the opposite needs of i) reducing the dimension of the CTMC as much as possible in order to make the solution efficient and fast, and ii) keeping the CTMC dimension large enough so that the truncated chain accurately approximates the original infinite one. In order to find a proper truncation criterion, we first discuss the marginal distributions of n_s and n_f.

Let us focus on the number of active sessions, n_s. Sessions are generated according to a Poisson process with rate λ_s. The lifetime of a session, i.e., the time a session spends in the system, is given by the sum of the interarrival times of the flows generated by the session. Interarrival times between flows of a session are i.i.d. negative exponential random variables X with rate λ_f. The lifetime Y of a session is thus given by,

$$Y = \sum_{i=0}^{\infty} \beta^i (1-\beta) X_i \tag{8}$$

where $\{X_i\}_{i \in \mathbb{N}}$ is a sequence of i.i.d. exponentially distributed random variables. It can be easily shown that Y is negative exponential distributed with mean value $E[Y] = E[X]\beta/(1-\beta) = \beta/[\lambda_f(1-\beta)]$. Since sessions are independent from each other, the evolution of the number of sessions in the system can be modeled by an M/M/∞ queue, whose arrival rate

[4] The operator $diags\{x_1, x_2, \cdots, x_n\}$ defines a diagonal matrix in $\mathbb{R}^{n \times n}$ whose elements along the diagonal are given by $x_1, x_2, \cdots, x_n$.

is λ_s and whose mean service time is $E[Y]$. It follows that the distribution of n_s is Poisson with parameter $\delta = \lambda_s E[Y] = \lambda_s \beta/[\lambda_f(1-\beta)]$.

We now consider the number of active flows, n_f. In order to model the flow arrival process, we number flows according to the order in which they are generated by the session they belong to: A flow of type i is the $i-$th flow generated by a session. We model the flow arrival process by the infinite queuing network reported in Fig. 6.a, where arrivals at queue i represent the arrivals of type i flows. The service time of a customer in queue i represents the interarrival time between the i-th flow and the $(i+1)-$th flow of a session and is distributed according to a negative exponential distribution with rate λ_f. Since type 1 flows are generated at the arrival of sessions, type 1 flows arrive at queue 1 according to a Poisson process with rate λ_s. A customer leaving queue 1 enters queue 2 with probability $(1-\beta)$, which is the probability that the flow was not the last one of the session. Since the departure process from an infinite queue with Poisson arrivals is Poisson too, the arrival process at queue 2 is Poisson with rate $(1-\beta)\lambda_s$. Thus, the arrival process at queue i is Poisson with rate $\beta^{i-1}\lambda_s$. This queuing network is equivalent to the queue with feedback shown in Fig. 6.b. Notice that, as is well known, despite the arrival processes at all queues in Fig. 6.a are Poisson, the infinite sum of the arrival processes which is the process entering the queue with feedback in Fig. 6.b is not Poisson.

Since the lifetime of flows in the system is negative exponential distributed with rate μ_f, the behavior of flows can be described by a set of infinite M/M/∞ queues, where the arrival rate at queue i is equal to $(1-\beta)^{i-1}\lambda_s$ and the average service time is given by $1/\mu_f$; or equivalently by a queue with feedback as the one in Fig. 6.b with μ_f instead of λ_f. It results that the distribution of n_f, the number of flows in the system, is Poisson distributed with rate $\lambda_s/(\beta\mu_f)$. Given that the marginal distributions of n_s and n_f is Poisson distributed, we can truncate the infinitesimal generator choosing (s_m, s_M) and (f_m, f_M) such that

$$\sum_{k \in \{s_m, \cdots, s_M\}} p_{n_s}(k) = 0.9 \tag{9}$$

$$\sum_{k \in \{f_m, \cdots, f_M\}} p_{n_f}(k) = 0.9 \tag{10}$$

where $p_{n_s}(k)$ and $p_{n_f}(k)$ are the probability density functions of n_s and n_f.

D. The Buffer Model (MMPP/M/1/m Queue)

For network planning and dimensioning, we are typically interested in the performance of a queue which represents the bottleneck of a network.

Besides the advantages of being simple to implement and efficient, a synthetic Markovian source as the one we propose has the additional advantage of allowing a Markovian model of a queue.

In general, the buffer model can be described by a MMPP/GI/1/m queue, where the service time represents the

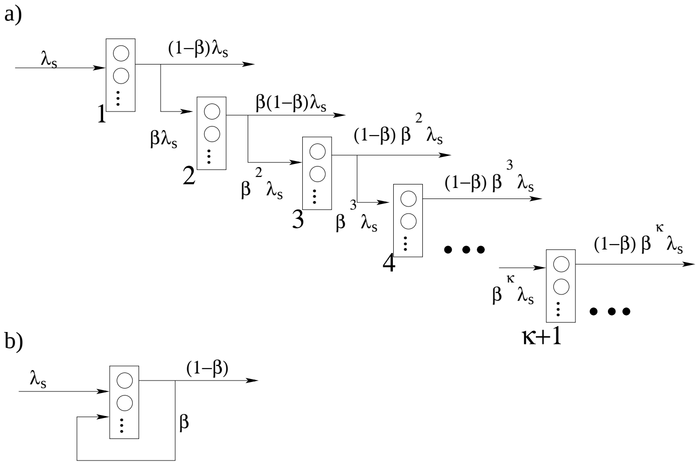

Fig. 6. Sessions, flows, and packets as seen by the model

transmission time of a packet, and can be easily derived from the capacity of the link and the distribution of the packet length. The general service time distribution can be approximated by a phase type distribution. However, for simplicity, we consider the case of exponentially distributed service time; we validate the exponential assumption in Section V and we discuss the extension to phase type service times at the end of this section.

By adopting an exponential service time distribution, we obtain an MMPP/M/1/m queuing system. The stochastic process which describes the dynamic of the system is a vector process $\{Z(t), t \in \mathbb{R}^+\}$, $Z(t) = \{R(t), J(t)\}$, where $R(t)$ denotes the number of packets in the queue, $J(t) = (n_f(t), n_s(t))$ denotes the phase of the modulating chain and is a vector Markov process too.

The state space of the considered Markov process is $S = \{z = (r, n_f, n_s) \in \mathbb{N}^3$ with $0 \le r \le m, f_m \le n_f \le f_M$ and $s_m \le n_s \le s_M\}$. The infinitesimal generator of such a CTMC is A,

$$A = \begin{pmatrix} A_{00} & A_0 & 0 & \cdots & \cdots & 0 \\ A_{10} & A_1 & A_0 & \ddots & \cdots & \vdots \\ 0 & A_2 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \cdots & \ddots & A_2 & A_1 & A_0 \\ 0 & \cdots & \cdots & 0 & A_2 & A_{mm} \end{pmatrix}$$

where $A_{00} = Q - \Lambda$, $A_0 = \Lambda$, $A_1 = Q - \mu I_n - \Lambda$, $A_{mm} = Q - \mu I_n$, $A_2 = \mu I_n$. $A_{00}, A_{10}, A_0, A_1, A_2, A_{mm}, I_n \in \mathbb{R}^{n \times n}$, and I_n is the identity matrix.

$Z(t)$ is a finite QBD (Quasi Birth Death) process whose solution has been studied in many papers, see for example [42], [43], [44], [45]. We adopt the solution proposed by [44], the Improved Logarithmic Reduction Algorithm (ILRA). Let $\rho = \pi \Lambda e/\mu$ and let the steady-state distribution be $\hat{\pi}$. We are interested in the buffer length distribution π^q_r which can be obtained from [44],

$$\pi^q_r = \hat{\pi}_r e = \lim_{t \to \infty} \mathbb{P}\{R(t) = r\} \quad r = 0, \cdots, m$$

$$\hat{\pi}_r = x_0\ R^r + x_m\ S^{m-r}[I_n - (SR)^r] \tag{11}$$

where R, S are two matrix geometric terms and (x_0, x_m) are the solution of the two boundary conditions of the QBD finite process:

$$\hat{A}^{[0,m]} = \begin{pmatrix} \hat{A}_{00}^{[0,m]} & \hat{A}_{0m}^{[0,m]} \\ \hat{A}_{m0}^{[0,m]} & \hat{A}_{mm}^{[0,m]} \end{pmatrix}$$

$$\begin{aligned}
\hat{A}_{00}^{[0,m]} &= Q - \Lambda + A_0 G(R) && (12)\\
\hat{A}_{0m}^{[0,m]} &= R^m A_0 [I_n - G(R)] && (13)\\
\hat{A}_{m0}^{[0,m]} &= S^{m-1}[I - SR]A_2 && (14)\\
\hat{A}_{mm}^{[0,m]} &= W(S) + A_0 - (SR)^m\ A_0[I_n - G(R)] && (15)\\
G(R) &= [-(A_1 + R\ A_2)^{-1}]A_2 \\
W(S) &= A_1 + S\ A_0
\end{aligned}$$

It can be shown that $\hat{A}^{[0,m]}$ is an infinitesimal generator of a CTMC with (x_0, x_m) as a steady-state vector,

$$(x_0, x_m)\hat{A}^{[0,m]} = 0, \ \ (x_0, x_m) \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} e = 1$$

$$\begin{aligned}
B_1 &= I_n + R \textstyle\sum_{r=0}^{m-1} R^r \\
B_2 &= \textstyle\sum_{r=0}^{m-1} S^r - S^m R \sum_{r=0}^{m-1} R^r
\end{aligned}$$

If $\rho < 1$, then $\sum_{r=0}^{m-1} R^r = (I_n - R)^{-1}(I_n - R^m)$, else, if $\rho > 1$, $\sum_{r=0}^{m-1} S^r = (I_n - S)^{-1}(I_n - S^m)$, and the boundary conditions can be simplified taking into account the load value. The two terms R and S are two matrix geometric terms which are the minimal nonnegative solution of the two equations:

$$0 = R^2 A_2 + RA_1 + A_0 \tag{16}$$

$$0 = S^2 A_0 + SA_1 + A_2 \tag{17}$$

Observe that (11) can be extended to the infinite buffer case, which is the well known matrix geometric solution,

$$\hat{\pi}_r = x_0\ R^r, \ \ r \geq\ 0 \tag{18}$$

with boundary conditions:

$$x_0 \hat{A}_{00}^{[0,m]} = 0$$

$$x_0[I - R]^{-1} e = 1$$

The solution of (11) and (18) becomes more and more difficult as the matrix dimension increases; thus, the use of efficient iterative methods is a must [46]. The solution of the finite buffer case is more complex than that of the infinite case, in terms of both memory requirements and computation time. Indeed, the finite case needs the computation of two matrix geometric terms (instead of one) and the solution of a linear system twice as large as in the infinite case. However, despite its complexity, the finite buffer case must be solved in many cases of practical interest as, for example, for dimensioning the buffer size, in which case we typically need to compute the loss probability due to buffer overflow. A nice feature of the method in [44] is that the loss probability due to the buffer overflow can be directly evaluated through the recursive formula,

$$\pi_m^{q,(m)} = \pi^t (I_n - V_m) \tag{19}$$

$$\begin{aligned}
\Psi &= Q - e\pi^t && (20)\\
V_1 &= I_n - [I_n + \mu(Q - \Lambda)^{-1} e\pi^t && (21)\\
&\quad + \mu(Q - \Lambda)^{-1} Q \Psi^{-1}]^{-1} && (22)\\
V_m &= A_2(-A_1^{-1})[I - V_{m-1} A_0 (-A_1^{-1})]^{-1} && (23)
\end{aligned}$$

This recursive formula is very useful because the computation of the loss probability for the case of buffer size equal to m provides, as a side product, the loss probability observed for all values of the buffer size smaller than m; thus, it is very efficient for network dimensioning.

As previously mentioned, the above problem can be easily extended to consider phase type distributions of the service time instead of exponential distributions. Such an extension requires the introduction of a random process component which accounts for the service phase. However, the increase of the dimension of the involved matrices makes the MMPP/Ph/1/m queue hard to solve in terms of time and memory requirements. Thus, in what follows we consider only exponentially distributed service times.

V. Performance Evaluation

The comparison of Table IV with Tables II and III indicates that the proposed MMPP model captures the LRD characteristics of the traffic we measured at the edge router interconnecting the Politecnico di Torino network to GARR/B-TEN. This is hardly a surprise, since we tuned the model to obtain this result. It however indicates that the proposed simple MMPP model is capable of exhibiting LRD behaviors over the time scales of interest.

In this section we further evaluate the performance of the proposed MMPP model in two ways: i) we study the behavior of the synthetic traffic produced by the MMPP model when feeding a buffer in front of a transmission link, and compare the results against those produced by measured traffic traces, ii) we investigate the predictability and tunability of the model when used as a synthetic source of aggregate Internet traffic, i.e., we discuss the model effectiveness in representing different traffic scenarios.

The analysis in this section considers the 2001 traces, for which the values of the measured parameters are reported in Table V.

A. Queuing Analysis

In order to evaluate the accuracy of the model as a synthetic traffic source, we consider a queuing system and we compare the performance obtained by feeding the queue with the synthetic traffic generated by the MMPP model with the results obtained from the real trace. For comparison purposes only, and in order to show the importance of introducing some memory in the input

TABLE V
MEASURED VALUES FROM REAL TRACES USED IN SETTING THE MODEL PARAMETERS

	Peak '01	Night '01
$\hat{N}_p$	22.03	14.35
$\hat{\lambda}_p$	29.57	352.9
$\hat{\Lambda}_f$	49.95	2.78
$\hat{\Lambda}_p$	1113	40.06

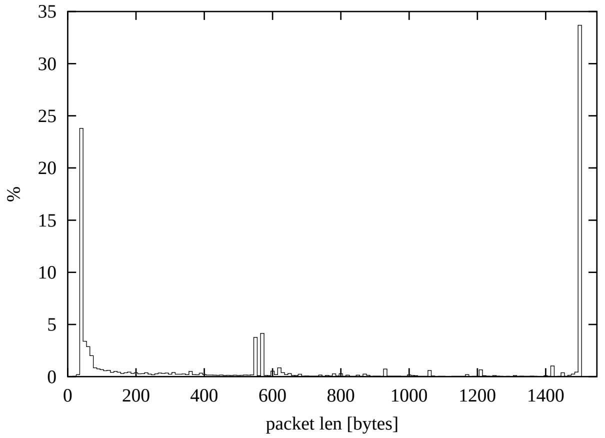

Fig. 7. Packet length distribution used in simulations

traffic model, we also plot the queue performance obtained when Poisson traffic feeds the queue.

We first consider the case of infinite buffer. The service time distribution reflects the packet length distribution measured from the real traces, which is reported in Fig. 7, and exhibits the well-known multi-mode behavior, with peaks for very short packets and for the different MTUs (Maximum Transfer Units) in the network, with a dominating peak at 1,500 bytes, due to the size of Ethernet frame. In order to evaluate the performance of the queue under different values of the load, we change the average service time while keeping the packet length distribution unchanged. Notice that the load of the queue has no relation with the actual load of the link where the traces were collected. The results for the synthetic traffic are obtained by simulating the corresponding MMPP/GI/1 queue.

Fig. 8 reports the queue length distribution for the Peak'01 trace. The thin dashed line is obtained by using as input the measured trace, the solid line is obtained with the MMPP model and the thick dashed line with a simple Poisson process whose rate matches the average packet arrival rate measured on the trace. Plots refer to four different loads: 0.9, 0.8, 0.7, and 0.6. For the real trace, the tail of the distributions below 10^{-4} becomes noisy due to lack of samples. The buffering behavior of the model matches quite well that of the measured traces, while, as expected, the Poisson model underestimates the buffer level of orders of magnitude. Notice that the accuracy of the MMPP model predictions tends to increase for large values of the load, which correspond to the most interesting cases for the system

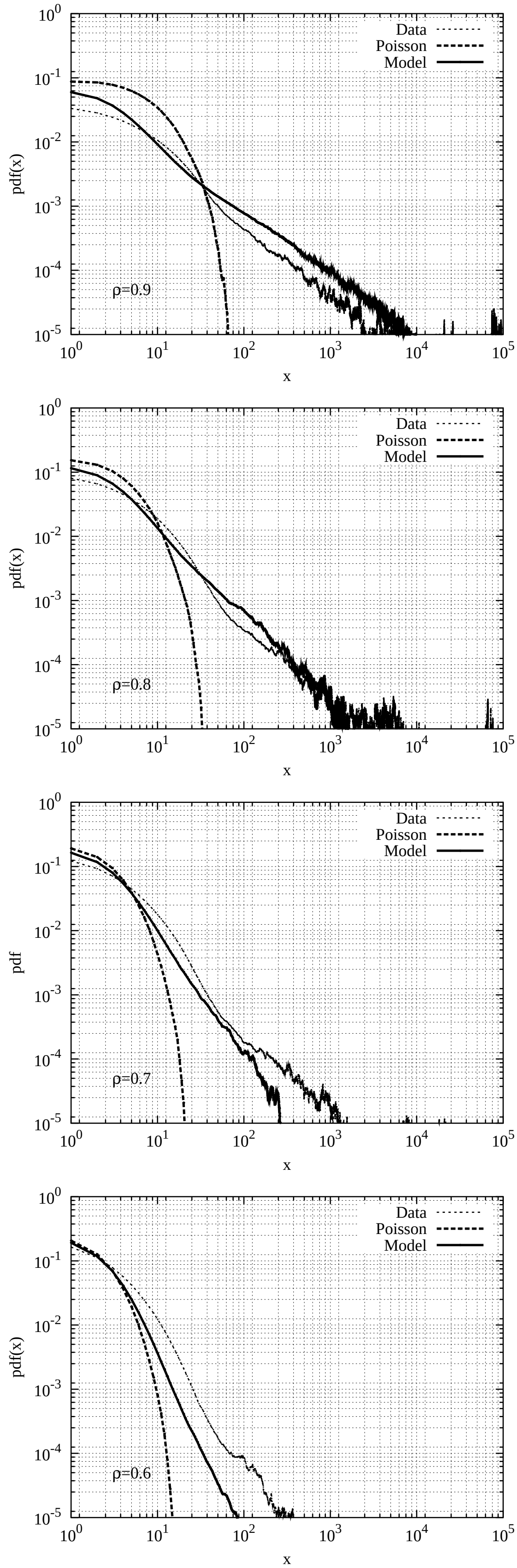

Fig. 8. Buffer occupancy distribution for the model, the Peak'01 trace and Poisson arrivals; load ρ = 0.9, 0.8, 0.7 and 0.6

designer.

Similar results are shown in Fig. 9 for the Night'01 trace. In this case, the accuracy of the model is less satisfactory, especially when the load is light. It must be noted, however, that this scenario is both less interesting and somewhat more "artificial" than the previous one. First of all, the number of points in the measured trace is about 50 times smaller than during peak hours (see Table I) and this explains why trace-driven curves are noisier. Second, the real link load at night is extremely low, thus artificially forcing the load to values higher than 0.6 significantly modifies the overall scenario. Yet, despite of this, the MMPP model matches quite well the tail of the distribution, which is typically the most crucial part of the curve.

We now look at the case of finite buffers. The results for the MMPP model are derived analytically, by assuming that service times are exponentially distributed. Fig. 10 reports the queue length distribution for a finite buffer queue driven by the same arrival process as in the previous scenarios for the Peak'01 trace, with ρ=0.9. The considered buffer sizes are $B = 32$, 64, 128, 256, 512 packets. Notice again the accuracy of the model in evaluating the queue performance. The curves for the Poisson traffic with different values of the buffer size are indistinguishable.

We now consider the typical dimensioning problem: the evaluation of the impact of the buffer size on the packet loss probability. Results are shown in Fig. 11. Two curves refer to the MMPP model. The solid line reports analytical results obtained with the assumption that service times are exponentially distributed; the markers are derived from the model by simulation using the same distribution of the service time as for the real trace. The figure proves that the impact of the exponential assumption for service times is marginal. Again, notice that analytical predictions are very accurate.

In terms of complexity, the numerical approach depends mainly on the size of the matrices which describe the Markov process. The computation of the steady-state distribution of the MMPP is efficient because the matrix Q is sparse, and finding the matrix geometrix terms is also efficient thank to the ILRA solution method, even for very large matrices. On the contrary, the computation of the boundaries can be a difficult task because it requires solving a dense linear system.

B. Sensitivity Analysis

We now discuss the impact of the MMPP model parameters on both the Hurst parameters and the queuing behavior. In particular, we focus on the effect of the average number of flows per session, N_f, which represents the long term memory of the model. We proceed as follows. We first derive the model parameters as described in Section IV-A. Then, we set a new value of N_f and, accordingly, we change λ_s so that the average number of flows generated in the time unit does not change. All the other model parameters are kept unchanged. Observe that the load is kept equal to the desired value (0.9) even if N_f is changed, as

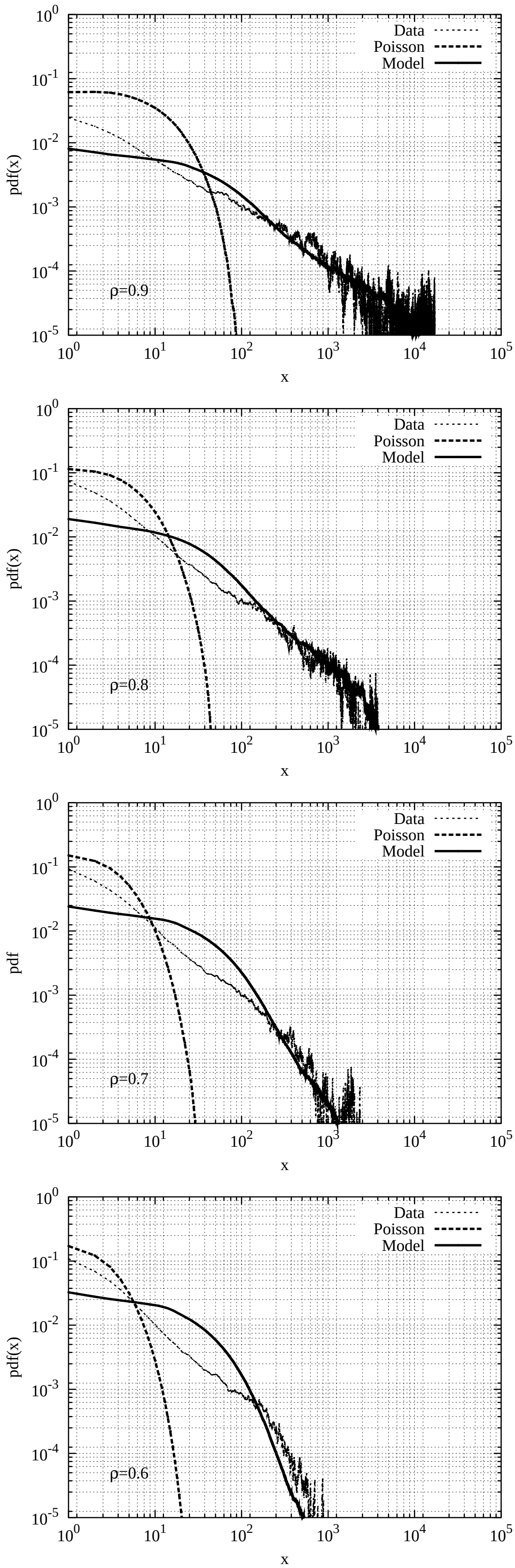

Fig. 9. Buffer occupancy distribution for the model, the Night'01 trace and Poisson arrivals; load ρ = 0.9, 0.8, 0.7 and 0.6

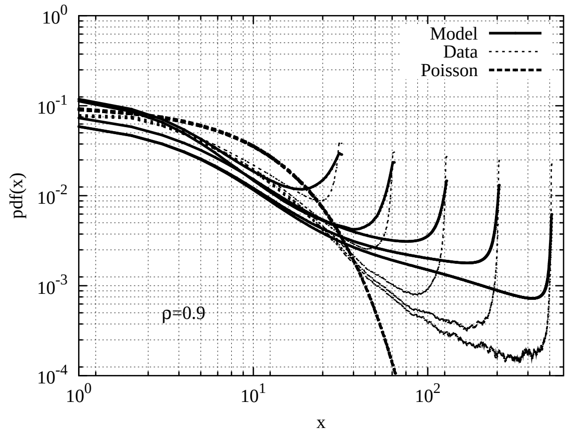

Fig. 10. Buffer occupancy distribution for the model, the Peak'01 trace and Poisson arrivals; finite buffer with capacity equal to 32, 64, 128, 256, 512 packets and load ρ = 0.9

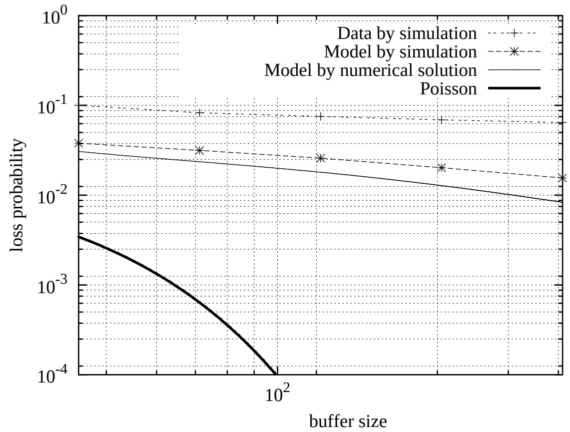

Fig. 11. Buffer loss probability for the model, the Peak'01 trace and Poisson arrivals; finite buffer with capacity equal from 32, 64, 128, 256, 512 packets and load ρ = 0.9

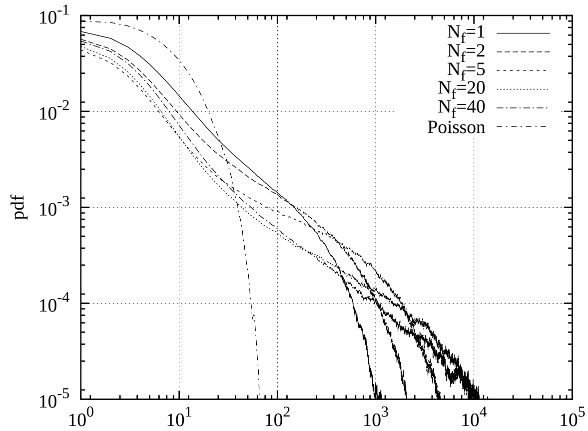

Fig. 12. Buffer occupancy distribution for the model with different values of N_f and λ_s. Offered load equal to 0.9

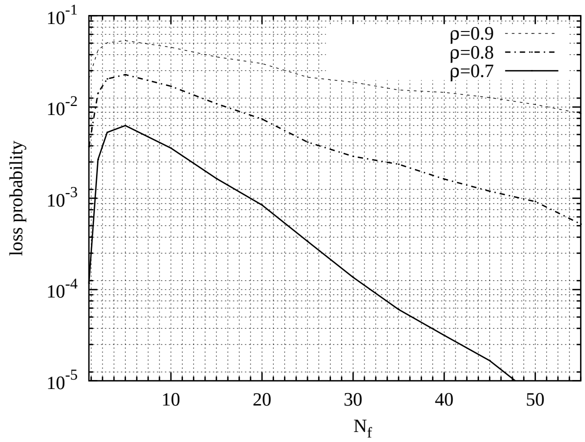

Fig. 13. Loss Probability with different values of N_f and N_p. Buffer size equal to 512.

shown by the following equations:

$$\begin{aligned}\mathbb{E}[\Lambda_f] &= \lambda_s + \lambda_f \mathbb{E}[n_s] = \lambda_s + \lambda_f \frac{\lambda_s}{\lambda_f}(N_f - 1) && (24)\\ &= \hat{\Lambda}_f && (25)\\ \mathbb{E}[\Lambda_p] &= \mathbb{E}[\Lambda_f]N_p = \hat{\Lambda}_f \hat{N}_p = \hat{\Lambda}_p && (26)\end{aligned}$$

This result is obvious, if we recall that N_f and C do not change the first order statistic but they act only on the second order statistics.

Results are shown in Fig. 12 for different values of N_f. Again, for comparison purposes only, we report the curve derived with Poisson packet arrivals. The case $N_f = 1$ corresponds to one flow only per session, which implies that the flow arrival process is Poisson. Increasing the number of flows per session makes the queue tail heavier: as N_f increases, the range where the queue decay follows roughly a power law becomes longer. Clearly, there is always a value beyond which the queue decay is exponential and this value increases with N_f. Indeed, the behavior of our model confirms the results obtained in [47], where the Hurst parameter of measured traffic, as well as the queuing behavior of the same traffic, are fitted through the use of phase type distributions.

Thanks to its simplicity and to the intuitive meaning of the parameters, the model can be effectively used as a manageable tool for IP network dimensioning and design. Indeed, having proved that the model accurately represents real traffic behavior, the model can be used to assess the performance of a system under variable traffic conditions, by simply changing the value of the parameters. Consider for example that for the network design we are interested in evaluating the impact of different traffic mixes on a finite buffer. We fix a value of N_f and modify accordingly N_p so that the average offered load to the buffer is constant. This corresponds to traffic mixes composed by either a small number of long flows or a large number of short flows. Fig. 13 plots the loss probability for a finite buffer of 512 packets, and for average load equal to $0.7,\ 0.8,\ 0.9$. The Figure shows that the loss probability initially increases for increasing values of N_f, but for values of N_f larger than about 5, the loss probability decreases. Indeed, the loss probability is strictly related to the correlation in the packet arrival process:

for N_f close to one (and therefore for large values of N_p), we expect less correlation in the traffic mix, having just one long flow per session, i.e., flows and sessions tend to coincide. Similarly, large values of N_f force small values of N_p, which leads to many one-packet long flows, i.e., flows and packets tend to coincide, thus reducing correlation.

VI. CONCLUSIONS

In this paper we have proposed a simple MMPP Internet traffic model that is capable of well approximating the traffic characteristics measured at the edge router of our institution. The model is based on a layered structure of sessions, that generate flows, that finally generate packets.

The characteristics of the synthetic traffic generated with the model match the LRD characteristics observed in the measured traces over the time scales of interest. One of the interesting features of the MMPP model is that it requires as inputs five parameters only. Three of these parameters can be directly mapped onto average traffic parameters, such as the average flow arrival rate, the average number of packets per flow, and the average arrival rate of packets within flows. The other two parameters define the notion of session, and are used to control the Hurst parameter of the synthetic traffic on the considered scaling range.

Most interesting is that the behavior of the synthetic traffic in a queue with either finite or infinite buffer matches very well the behavior of the measured traces. Thus, the proposed MMPP model can be considered an accurate descriptor of aggregate Internet traffic, and can be effectively used to dimension buffer sizes and link capacities.

The key features of the proposed MMPP model are its simplicity and its intuitive structure. While, on the one hand, these features allow an accurate match of the characteristics of measured traffic, on the other hand, they allow the model to be used by traffic engineers with only limited knowledge of the sophisticated theoretical aspects of LRD processes.

REFERENCES

[1] W.E. Leland, M.S. Taqqu, W. Willinger, V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended version)," *IEEE/ACM Transaction on Networking*, Vol.2, N. 1, pp. 1–15, Jan. 1994.

[2] V. Paxson, S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking*, Vol.3, N.3, pp. 226–244, Jun. 1995.

[3] M.E. Crovella, A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Transaction on Networking*, Vol.5, N.6, pp. 835–846, Dec. 1997.

[4] W. Willinger, M.S. Taqqu, R. Sherman, D.V. Wilson, "Self-Similarity Through High Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level," *IEEE/ACM Transaction on Networking*, Vol.5, N.1, pp. 71–86, Jan. 1997.

[5] M. E. Crovella, M. S. Taqqu, A. Bestavros, "Heavy-Tailed Probability Distributions in the World Wide Web," In *A Practical Guide To Heavy Tails*, Chapter 1, Chapman & Hall, New York, pp. 3–26, 1998.

[6] D. R. Cox, "Long-range dependence: A review," *Statistics: An Appraisal*, H. A. David and H. T. David, Eds. Ames, IA: Iowa State Univ. Press, pp.55-74, 1984.

[7] A. Feldmann, A. Gilbert, W. Willinger and T. Kurtz, *The Changing Nature of Network Traffic: Scaling Phenomena*, Computer Communication Review 28, No.2, Ap. 1998

[8] A. Erramilli, O. Narayan and W. Willinger, *Experimental Queueing Analysis with Long-Range Dependent Packet Traffic*, IEEE/ACM Transactions on Networking, Vol.4, No.2, pp.209–223, 1996

[9] A. Veres, M. Boda, "The Chaotic Nature of TCP Congestion Control," *IEEE INFOCOM 2000*, Tel Aviv, Israel, pp. 1715-1723, Mar. 2000.

[10] A. Veres, Z. Kenesi, S. Molnar, G. Vattay, "On the propagation of long-range dependence in the Internet," *ACM Sigcomm 2000*, Stockholm, Sweden, pp. 243-254, Sept. 2000.

[11] D. R. Figueiredo, B. Liu, V. Misra, and D. Towsley. *On the Autocorrelation Structure of TCP Traffic.* Technical Report UMass-CMPSC-00-55, University of Massachusetts, Amherst, Computer Science Department, November 2000.

[12] I. Norros, *On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks*, IEEE Journal on Selected Areas in Communications, Vol.13, pp.953–962, Aug. 1995

[13] I. Norros, *A storage model with self-similar input.*, Queueing Systems, 16:387–396, 1994.

[14] H. Riedi, M.S. Crouse, V.J. Ribeiro, R.G. Baraniuk, "A multifractal Wavelet Model with application to data traffic," *IEEE Transaction on Information Theory*, Vol. 45, No. 4, pp. 992-1018, Apr. 1999.

[15] M. Taqqu and V. Teverosky, *Is Network Traffic Self-Similar or Multifractal ?*, Fractals, Vol.5, No.1, pp.63–73, 1997

[16] R.H. Riedi and J. Lévy Véhel, *Multifractal Properties of TCP Traffic: a Numerical Study*, IEEE Transactions on Networking, Nov. 1997 (Extended version appeared as INRIA research report 3129, Mar. 1997)

[17] S.Roux, D.Veitch, P.Abry, L.Huang, P.Flandrin and J.Micheel, *Statistical Scaling Analysis of TCP/IP Data*, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01), Special session: Network Inference and Traffic Modeling, May 2001

[18] A. Feldmann, A.C. Gilbert, W. Willinger, "Data Networks as Cascades: Investigating the Multifractsl Nature of Internet WAN Traffic," *ACM Sigcomm*, Boston, Ma, pp. 42-55, Sept. 1998.

[19] D. Veitch, P. Abry, P. Flandrin and P. Chainais, *Infinitely Divisible Cascade Analysis of Network Traffic Data*, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'00), 2000.

[20] S. Ma, C. Ji, "Modeling Heterogeneous Network Traffic in Wavelet Domain", *IEEE/ACM Transaction on Networking*, Vol.9, No.5, pp. 634-649, Oct. 2001.

[21] A. De Vendictis, A. Baiocchi, *Wavelet Based Synthetic Generation of Internet Packet Delays*, in Proc. of ITC17, Salvador (Brazil), December 2001.

[22] A. Eramilli, R.P. Singh, "An Application of Deterministic Chaotic Maps to Model Packet Traffic," *Queueing Systems*, Vol.20, pp. 171–206, 1995.

[23] R.J. Mondragon, J.M. Pitts, D.K. Arrowsmith, "Chaotic Intermittency-Sawtooth Map Model of Aggregated Self-Similar Traffic Streams," *Electronic Letters*, Vol.36, N.2, pp. 184–186, 2000.

[24] M. Krunz, A. Makowski, "A Source Model for VBR Video Traffic Based on M/G/infinity Input Processes," IEEE Infocom 98, San Francisco, CA, USA, pp. 1441-1448, Apr. 1998.

[25] A. Horváth and M. Telek, *A Markovian Point Process Exhibiting Multifractal Behavior and its Application to Traffic Modeling*, 4th International Conference on Matrix-Analytic Methods in Stochastic Models, Adelaide (Australia), Jul. 2002

[26] A. T. Andersen, B. F. Nielsen, "A Markovian approach for modeling packet traffic with long-range dependence,"*IEEE journal on selected areas in communications*, Vol.16, No.5, pp. 719-732, 1998.

[27] S. Robert and J.Y. Le Boudec, *New Models for Pseudo Self-Similar Traffic*, Performance Evaluation, Vol.30, No.1-2, pp.57–68, 1997.

[28] S. Robert and J.Y. Le Boudec, *Can Self-Similar Traffic be Modeled by Markovian Process ?*, International Zurich Seminar on Digital Communication, Feb. 1996

[29] A. Reyes Lecuona, E. González Parada, E. Casilari, J.C. Casasola and A. Díaz Estrella, *A Page-Oriented WWW Traffic Model for Wireless System Simulations*, ITC16, Edinburgh, Jun. 1999

[30] M. Grosslauser, J. Bolot, "On the Relevance of Long-Range Dependencies in Network Traffic," *IEEE/ACM Transaction on Networking*, Vol.7, N.5, pp. 629–640, Oct. 1999.

[31] S. Ben Fredj, T. Bonald, A. Proutiere, G. Régnié, J. Roberts, "Statistical Bandwidth Sharing: a Study of Congestion at Flow Level," *Proc of Sigcomm 2001*, San Diego, CA, pp. 111-122, Aug. 2001.

[32] T. Bonald, A. Proutiere, G. Régnié, J. Roberts, "Insensitivity Results in Statistical Bandwidth Sharing," *International Teletraffic Conference (ITC) 2001*, San Paolo, Brasil, Nov. 2001.

[33] "GARR - Rete dell'Università e della Ricerca Scientifica Italiana," http://www.garr.it, 2002.

[34] S. McCanne, C. Leres, V. Jacobson, "Tcpdump," http://www.tcpdump.org, 2002.

[35] M. Mellia, A. Carpani, R. Lo Cigno, "Tstat web page," http://tstat.tlc.polito.it/, 2002.

[36] M. Mellia, A. Carpani, R. Lo Cigno, "Measuring TCP and IP Behavior on a Edge Node," *IEEE Globecom 2002*, Taipei, Nov. 2002.

[37] M.S. Taqqu, *Fractional Brownian Motion and Long Range Dependence,* Theory and Application of Long-Range Dependence P.Doukhan, G. Oppenheim, M.S. Taqqu Editors 2002.

[38] P. Abry, P. Flandrin, M.S. Taqqu and D.Veitch, *Self-similarity and Long-Range Dependence Through the Wavelet Lens*, In *Long Range Dependence: Theory and Applications*, Doukhan, Oppenheim, Taqqu Editors 2002

[39] P. Abry, D. Veitch, "Wavelet Analysis of Long Range Dependent Traffic," it IEEE Transactions on Information Theory, Vol.44, No.1 pp. 2-15, Jan. 1998.

[40] "Darryl Veitch Home Page," http://www.emulab.ee.mu.oz.au/ darryl/

[41] W. Fisher and K. Meier-Hellstern, *The Markov-modulated Poisson process (MMPP) cookbook* Performance Evaluation 18 (1992) 149-171

[42] M. F. Neuts, *Structured Stochastic Matrices of M/G/1 Type And Their Applications* Marcel Dekker, New York, 1989

[43] G.Latouche, V. Ramaswami, *A logarithmic Reduction Algorithm for quasi-birth-death processes,* J. App. Prob, 30 (1993), pp 650-674.

[44] U.R. Krieger, V, Naoumov and D. Wagner, *Analysis Of A Finite Buffer In An Advanced Packet-Switched Network* Special Issue on ATM Traffic Control and Performance Evaluation IEICE Transaction On Communication VolE00-B No.5, May 1998

[45] D. Bini, B. Meini, *On cyclic reduction applied to a class of Toeplitz-like matrices arising in queueing problems.* In W. J. Stewart, editor, Computations with Markov Chains, pages 21–38. Kluwer Academic Publisher, Boston, 1995.

[46] R. Barret, M. Berry, T.F. Chan, J.Demmel, J.M. Donato, J. Dongarra, V. Eijkhout, R.Pozo, C. Romine, H. Van der Vorst, *Templates for the solution of Linear Systems: Building Blocks for Iterative Methods* SIAM, 1994, Philapelphia, PA, book and code available at http://www.netlib.org/templates/Templates.html

[47] A. Horváth, G.I. Rózsa, M. Telek, "A map fitting method to approximate real traffic behaviour," *8-th IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks*, Ilkley, U.K., July 2000.