Catching IP traffic burstiness with a lightweight generator

Chloé Rolland[1], Julien Ridoux[2], and Bruno Baynat[1]

[1] Université Pierre et Marie Curie - Paris VI, LIP6/CNRS, UMR 7606, Paris, France
{rolland, baynat}@rp.lip6.fr
[2] ARC Special Research Center for Ultra-Broadband Information Networks (CUBIN), an affiliated program of National ICT Australia.
The University of Melbourne, Australia
j.ridoux@ee.unimelb.edu.au

Abstract. This paper presents LiTGen, an easy to use and tune open-loop traffic generator that statistically models IP traffic on a per user and application basis. From a packet level capture originating in an ISP wireless network[1], and taking the example of Web traffic, we show that the simple underlying structure of LiTGen is sufficient to reproduce the traffic burstiness accurately. In addition, the flexibility of LiTGen enables us to investigate the sensitivity of the traffic structure with respect to the distributions of the random variables involved in the model, and their possible dependencies.
Key words: traffic generator, scaling behaviors, second-order analysis, semi-experiments

1 Introduction

Measuring, understanding and reproducing network traffic characteristics are essential steps of traffic engineering. Traffic generators usually tackle the later step. Among past proposals, focusing primarily on Web traffic, [1] and [2] proposed hierarchical models, but did not validate them against real traffic traces. The work presented in [3] is an effort to generate representative traffic for multiple and independent applications. However, the model underlying the introduced generator was not designed to specify the packet level dynamics neither to catch the traffic scaling structure. Recently, [4] aims at reproducing the burstiness observed in captured traffic but relies on a third party link and network layers opaque emulator requiring high computing resources.

This paper presents LiTGen, a "**Li**ght **T**raffic **Gen**erator". LiTGen relies on a simple hierarchical description of traffic entities, most of them modeled by uncorrelated random variables and renewal processes. This design does not require to consider network or protocol characteristics (*e.g.* RTT, link capacities,

[1] This study would not have been conducted without the support of Sprint Labs. The authors would like to thank Sprint Labs for providing the wireless traces and particularly A. Sridharan for his support.

TCP dynamics...) and allows fast computation executed on a commonplace computer. Focusing on Web traffic, this paper confronts LiTGen to real traces captured on an operational wireless access network at Sprint Labs. Using a second-order analysis, we identify the dependencies across the random variables composing LiTGen's underlying model and prove LiTGen's ability to reproduce accurately the captured traffic and its properties over a wide range of timescales. To the best of our knowledge, we are the first ones to produce synthetic wireless traffic for the first two orders of the internal time series.

In the rest of this paper, section 2 describes LiTGen; section 3 develops the second-order analysis we used. We then investigate in section 4, the sensitivity of the traffic structure with respect to the distributions of the random variables involved in the underlying model. Finally we conclude this paper with a summary of our findings and directions for future work.

2 Building a lightweight traffic generator

2.1 Underlying Model

Earlier works identified several possible causes of correlation in IP traffic, namely the superimposition of traffic sources modeled by heavy tailed distributions [5, 6] and the inherent structure and interactions of protocol layers [7]. These assumptions call on the conception of a traffic generator based on a user-oriented approach and a semantically meaningful hierarchical model. This model is made of several levels, each of them characterized by a specific traffic entity. Each network entity defined is represented by one or several random variables either related to a time (duration or inter-arrival time) or a size metric.

Session level. We assume each user undergoes an infinite succession of **session** and inter-session periods. Taking the example of Web traffic, a user downloads a certain number of Web pages during a session. The random variable $N_{session}$ describes the user's session size, counting the number of pages downloaded, while T_{IS} characterizes the inter-session durations.

Page level. The Web **pages** downloaded during a session are separated by reading times (OFF periods). We define two random variables to characterize this level: the page size N_{page} describes the number of objects involved in a page and T_{off} models the corresponding reading duration.

Object level. Each page is split up into a set of requests (sent by the user) and responses (from the server), where responses gather the page's **objects** (HTML skeletons or embedded objects such as pictures). IA_{obj} and N_{obj} characterize respectively the objects inter-arrivals in a page and the number of packets in an object.

Packet level. Finally, each object is made of a set of **packets**. IA_{pkt} characterizes the successive inter-arrivals times between packets in an object.

LiTGen's model is then kept simple since not modeling the client/server interactions. Also the model does not rely on a complex emulator that would reproduce the link layer or TCP dynamics. Note however that network characteristics and/or TCP dynamics can explicitly be taken into account by introducing

simple queuing or Markovian models as input of our traffic generator. This work is currently under investigation. Finally, note that one can equivalently remove from the hierarchy the session level by including the inter-sessions durations in the OFF periods durations distribution. Nevertheless, it would make the characterization of T_{off} more complex and LiTGen less easy to use in practice.

2.2 Traffic entities identification

To calibrate LiTGen's underlying model, we benefit from data traces captured on the Sprint PCS CDMA-1xRTT access network. The traffic has been captured on an OC-3 collecting link and corresponds to tens of wireless access cells (more details about the trace and its differences with wireline traffic, for both upload and download traffic, can be found in [8]). The trace consists of a collection of IP packets with accurate timestamps, entire TCP/IP headers, and provides a large diversity of users' applications traffic. Because it has been extensively studied and is well known, we focus in this paper on the Web traffic downloaded by the mobile users on the wireless network to validate LiTGen. The application of LiTGen to other types of traffic, such as mail or P2P, can be found in [9].

The model calibration requires to identify the traffic entities from the captured trace. We aggregate packets to identify objects, pages and sessions, based on the 5-tuple associated to each packet ($\{IP_D$, IP_S, $port_D$, $port_S$, proto$\}$). A filter based on a source port number selection ($\{80, 8080, 443\}$) retains Web packets only[2]. A user's Web packets share the same destination IP address and are then grouped into sets of a given ($\{IP_S, port_D\}$) pair. These sets correspond to the Web flows the user requested.

We identify objects within the packets sets by analyzing the TCP flags (SYN, ACK...) of the TCP/IP headers. This method, comparable to [10], is particularly useful when HTTP persistent connections are used since a set of packets may carry several Web objects.

Because we do not have access to the packets' payloads, the identification of Web pages relies on heuristics to determine their boundaries. Once Web objects have been delimited, we define *active periods* during which one or several objects are being downloaded, opposed to *inactive periods*. The silence corresponding to a given inactive period may be due to the user (thus it is an OFF period) or the system (*e.g.* idle time due to the Web browser). We use a temporal clustering method (also used in [1, 2, 10]) to distinguish those two kinds of silences: an inactive period that lasts for more than a predefined threshold is labeled as an OFF period. Based on the inactive periods distribution, we empirically set the threshold to 1 second, a result consistent with [1, 2, 10].

Similarly, we aggregate Web pages into user sessions, setting empirically a threshold value to 300 seconds to separate OFF periods from inter-sessions[3].

[2] See [9] for other kinds of traffic.

[3] Note that the precise value of this threshold does not impact significantly the results.

2.3 Traffic Generation

While in this paper we apply LiTGen to Web traffic, it can obviously be adapted to any kind of traffic [9]. When numerous applications are multiplexed, we first set the number of users for each of them. For validation purposes we extract each application proportion and number of users from the captured trace. In an operational network these statistics can be derived from operator's knowledge of customers subscribed services. LiTGen then generates traffic for each user independently, from upper level entities (sessions) to lower ones (packets). For a given user, the process begins by generating a session starting at time $t = 0$. Lower levels traffic entities are then created until all the user's packets have been generated. A random circular shift is performed on each packet timestamp to accurately mix the different users traffic, since the final synthetic trace is obtained by superimposing synthetic traffic of all users and all applications.

3 Validation

We evaluate LiTGen on its ability to capture the complexity of the traffic correlation structure. To this aim, we use an energy spectrum comparison method to match the packets arrivals time series extracted from the captured trace and the corresponding synthetic trace. Since the 24-hour trace is not stationary (see [8] for details), the analysis is performed on a one-hour long period extracted from the entire captured trace. The results presented in the following correspond then to a given one-hour period; similar results were obtained for the other one-hour extracted traces.

3.1 Wavelet analysis

We use the Logscale Diagram Estimate or LDE [11] to perform discrete wavelet transform analysis. For a given time series of packets arrivals, the LDE produces a logarithm plot of the data wavelet spectrum estimates. Although the LDE has the ability to identify correlation structures in the data trace [12], we mainly use it to assess the accuracy of the synthetic traces produced by LiTGen.

We first generate synthetic traffic using a simple version of our generator, called *basic* LiTGen. In this version, all traffic entities are generated from renewal processes, using the empirical distributions extracted from the captured trace; and no dependency of any kind is introduced between the random variables. Figure 1 shows that the spectrum of the synthetic trace produced by basic LiTGen (thin curve) is far different from the captured trace spectrum. As a first conclusion basic LiTGen does not succeed in reproducing the captured traffic scaling structure with a good accuracy. We thus need to introduce additional dependencies between the random variables. In the following we investigate the impact of the introduction of a very simple correlation structure in LiTGen on the produced traces.

Previous studies [8, 13] pointed out that a great part of the LDE energy was due to the organization of packets within flows. This leads to refine LiTGen's

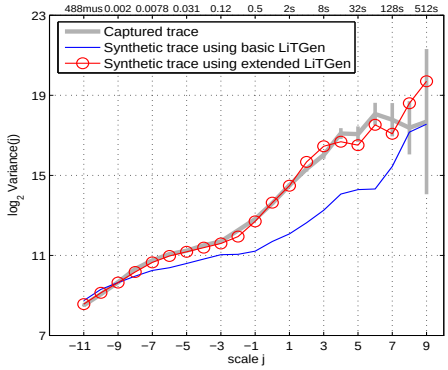


Fig. 1. Evaluating basic VS extended LiTGen

Approximated $f(s) = mean(IA^s_{pkt})$	
$\widehat{f(s)} = a.s^b$, $a = 0.8811$ and $b = -0.5897$	
Indices of Goodness of fit	
Sum of Squares due to Error	0.1868
Square of the multiple correlation (R-Square)	0.8712
Degrees of Freedom Adjusted R-Square	0.8693
Root Mean Squared Error	0.0524

Table 1. Fitting the average values of IA^s_{pkt}.

model: in this extension, referred to as *extended* LiTGen, the in-objects packets arrival times are still modeled by renewal processes, but now the average in-object packets inter-arrival times depend on the corresponding object size. Thus, in order to evaluate extended LiTGen, we extract size-dependent empirical distributions of in-objects packets inter-arrivals from the captured trace (the maximum object's size in terms of packets, extracted from the captured trace, is equal to 15547). When generating traffic, the packets inter-arrivals in a Web object of size s are then taken from the corresponding IA^s_{pkt} distribution.

The resulting spectrum (circle curve in Fig. 1) is barely distinguishable from the captured one. The introduction of this simple dependency between the objects sizes and the packets inter-arrival times succeeds in reproducing accurately the traffic correlation structure. This dependency may reflect the impact of TCP slow start on objects of different sizes: the bigger an object, the shorter its average packets inter-arrival times. Indeed, the average values of the IA^s_{pkt} distributions can be approximated with a good accuracy by a decreasing power law, strengthening the previous conjecture. Table 1 gives the goodness of such a fit taking into account the objects of size $s < 70$ (fit with cutoff, the population of objects of larger size is scarce, 99% of the captured values correspond to objects of size $s < 12$).

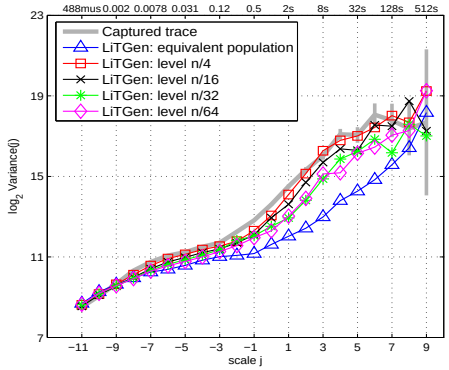
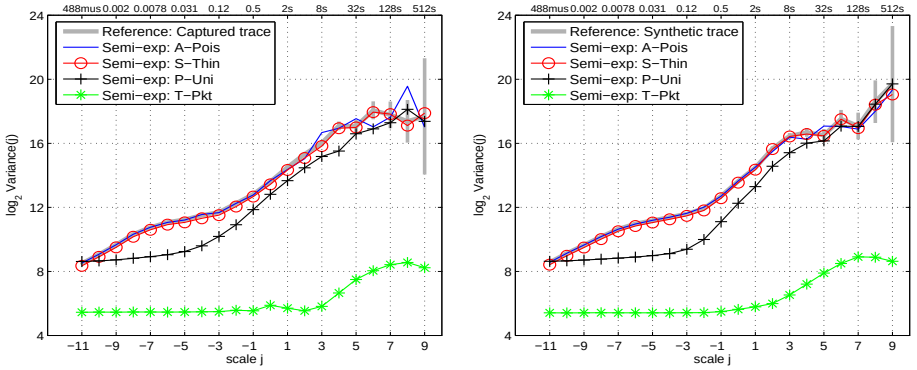


Fig. 2. Investigating grouped IA^g_{pkt} distributions in LiTGen

Without considering network or protocol peculiarities, extended LiTGen reproduces the second order traffic characteristics while remaining much simpler than [4]. However, in order to use extended LiTGen in operational conditions, we need to be able to characterize how the packets inter-arrival times distribution depends on the objects sizes. Obviously, the best would be to model analytically this relation, either by finding a suitable distribution whose parameters are described as functions of the objects sizes, or, as stated before, by involving simple (e.g. Markovian) TCP and/or network models as an input of our traffic generator. This work is currently under investigation.

As a first attempt to understand the impact of this relation between the in-objects packets inter-arrivals and the objects sizes over the traffic characteristics, and to help us carrying through analytical modeling, we first investigate the possibility of reducing the number of IA^s_{pkt} distributions to be considered, in order for LiTGen to remain simple and accurate. To this aim, we group objects of different sizes and compute "grouped" IA^g_{pkt} distributions. This operation requires to determine how many groups are needed and their composition.

We first group objects by maintaining an equivalent size p of the population for each group. The first group contains all the objects of size $s = 2$ and defines the population p. The following group is made by gathering objects of increasing sizes till the population of the group reaches p; and so on for the other groups. Because the population of the first group is large, we obtain only three groups corresponding to the following object sizes: {[2], [3;6], [7;15547]}. Thus, the packets inter-arrival times in a Web object composed of two packets are taken from the empirical distribution $IA^{s=2}_{pkt}$. The packets inter-arrival times in a Web object composed of three to six packets are taken from the empirical distribution $IA^{s=3\cup4\cup5\cup6}_{pkt}$; and so on. The resulting spectrum (triangle line in Fig. 2) is very similar to the basic LiTGen one. Finally, the small objects do not have a major impact on the spectrum. The large ones, even if less represented, carry a great number of packets, and have a significant impact on the spectra, at medium and large time scales.



(a) SE - captured trace

(b) SE - synthetic trace using extended LiTGen

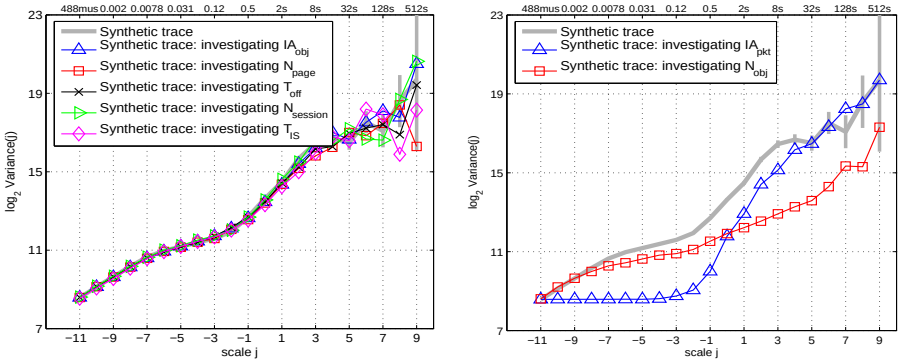
Fig. 3. LiTGen underlying model evaluation: semi-experiments methodology

In another attempt, we group the objects by describing the objects sizes distribution along a binary tree. The root of the tree is a single group corresponding to basic LiTGen and the use of one empirical distribution IA_{pkt} (not dependent on the objects sizes). The set of groups representing the tree leaves (deepest level n in the tree) corresponds then to extended LiTGen and the use of the maximum number n of empirical distributions IA^s_{pkt}, for each different object size s. Finally, the intermediate levels i in the tree correspond to the use of i empirical distributions $IA^{g_i}_{pkt}$ during the generation process. In figure 2, we observe spectra of the groups formed by specific levels of the binary tree. As the level observed is deeper in the tree (see the figure, from level $n/64$ to $n/4$), and the number of IA^g_{pkt} distributions used is greater, the corresponding spectrum gradually gets closer to the reference spectrum (and extended LiTGen) confirming the equivalent contribution of small and large objects. Other methods to build objects groups led to similar spectra and interpretation. In conclusion, no key value defining the number of groups appears; the ability to reduce the number of IA^s_{pkt} distributions to model depends directly on the desired accuracy.

3.2 Semi-experiments analysis

To exhibit the internal properties of LiTGen's synthetic traffic, we now conduct an analysis based on semi-experiments (SEs). SEs have been introduced in [14] and consist in an arbitrary but insightful manipulation of internal parameters of the time series studied. The comparison of the energy spectrum on the LDE before and after the SE leads to conclusions about the importance of the role played by the parameters modified by the SE. Using extended LiTGen, we apply the same set of SEs to the captured and synthetic traces and observe how they impact both spectra (Fig. 3(a) and 3(b)).

T-Pkt is a **T**runcation manipulation that transforms the objects arrival process by keeping only the first packet of each object. Removing packets decreases



(a) Insensitive: IA_{obj}, N_{page}, T_{off}, $N_{session}$ and T_{IS}

(b) Sensitive: IA_{pkt} and N_{obj}

Fig. 4. Sensitivity of the traffic scaling behaviors with regard to the r.v. distributions

the energy of the spectrum that takes smaller values. As shown in figures 3(a) and 3(b), **T-Pkt** has a similar impact on the captured and the synthetic traces.

S-Thin tests for the independence of objects. It randomly **S**elects objects with some probability, here equal to 0.9. When applying **S-Thin**, the spectra of the two traces keep the same shape but drop by a small amount close to $log_2(0.9) = -0.15$ (barely visible on the plots).

A-Pois targets the interactions between objects. This manipulation repositions the objects **A**rrival times according to a Poisson process and randomly permutes the objects order (preserving the internal structure of objects). While **A-Pois** is a drastic manipulation, it has similarly very little effect on the spectra of the two traces, indicating the negligible contribution of the objects arrival process.

P-Uni reveals the impact of in-objects packets burstiness. It uniformly distributes arrival times of packets in each object while preserving packets count and object duration. This manipulation flattens the spectrum from scales $j = -11$ to $j = -5$ in a comparable manner for the captured and synthetic traces.

To sum up, the captured and synthetic traces spectra present similar reactions to each SE. As a consequence, LiTGen reproduces the key internal properties of the captured traffic highlighted by the SEs, *i.e.* the objects arrival process has few influence; the objects can be considered as independent and the packets arrival process within objects contributes mostly to the energy spectrum. The simple structure of LiTGen, which still relies on renewal processes, is thus sufficient to reproduce these traffic internal properties.

4 Sensitivity of the traffic with regard to the distributions

We now investigate if "well-known" distributions can accurately approximate the empirical ones. Using statistical goodness of fit tests, we found that heavy

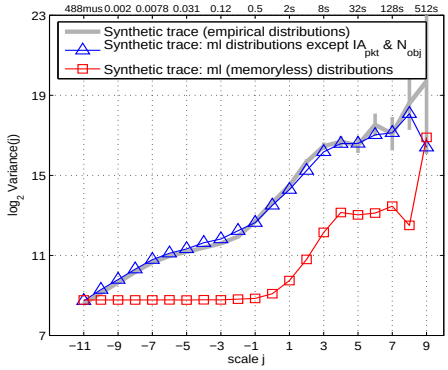


Fig. 5. Investigating the substitution of all empirical distributions

tailed distributions approximate well the majority of the random variables: while N_{page}, N_{obj} and T_{off} are close to power laws, $N_{session}$ and IA_{obj} are close to sub exponential distributions (respectively Lognormal and Weibull, see [15] for more details). Even if we did not manage to model IA_{pkt} by a well-known distribution accurately, it is closer to sub exponential distributions than exponential ones. Finally, T_{IS} can be well approximated by an exponential distribution.

While several works studied the relationship between traffic burstiness, and network or protocol characteristics (*e.g.* loss probabilities, RTT, link capacities, TCP dynamics) [4, 13, 16], the flexibility of extended LiTGen enables us to investigate the impact of random variables distributions on traffic burstiness. To this aim, we replace individually the empirical distribution of each random variable by a memoryless distribution (exponential or geometric) of same mean. We thus create seven synthetic traces, each one corresponding to a given random variable being replaced. We then compare these traces to the reference synthetic trace generated by extended LiTGen calibrated with the empirical distributions.

Fig. 4(a) illustrates the investigation of IA_{obj}, N_{page}, T_{off}, $N_{session}$ and T_{IS}. Even if heavy tailed distributions fit accurately these random variables, except T_{IS}, their modeling by memoryless distributions leads to very few impact on the data spectra. Indeed, we can barely distinguish the synthetic traces from the reference one. In conclusion, the traffic scaling structure of the studied wireless trace is insensitive with respect to the distributions of these random variables.

In Fig. 4(b), we present the investigation of IA_{pkt} and N_{obj}. As an example, we observe that modeling IA_{pkt} by an exponential distribution flattens the spectrum at scales below $j = -3$. The traffic scaling structure is then very sensitive to the distributions of IA_{pkt} and N_{obj}. This confirms the results obtained by the SEs methodology that highlighted the packets inter-arrival process within objects as the main source of energy in the spectrum.

Finally, we create two new synthetic traces, which we compare to the reference one. We obtain the first trace by modeling the *insensitive* random variables (cf. Fig. 4(a)) with memoryless distributions. We thus create a synthetic trace

in which only N_{obj} and IA_{pkt} are calibrated with empirical distributions. Fig. 5 shows that the corresponding spectrum (triangle line) matches the reference one. This result has a strong practical appeal: five of the seven random variables are modeled by memoryless distributions, while reproducing the traffic scaling structure. Moreover, we found that this spectrum reproduces the traffic internal properties highlighted by the semi-experiments in section 3.2. In Fig. 5, the last synthetic trace is obtained by modeling *all* the random variables with memoryless distributions. We observe a great deviation between the corresponding spectrum (square line) and the reference one, which confirms the sensitivity of the traffic with respect to the distributions of IA_{pkt} and N_{obj}.

5 Conclusion

This paper describes LiTGen, a light traffic generator that reproduces accurately the traffic scaling properties at small and large time scales. Illustrated on Web traffic, we show the accuracy of LiTGen to maintain second-order traffic characteristics without considering network or protocol peculiarities. We highlighted the dependency between objects sizes and in-object packets inter-arrivals, and showed how this impacts the quality of the generator.

Thanks to LiTGen, we investigated the impact of the random variables distributions describing the IP traffic structure. This investigation helped us to model simply these random variables and also identified the crucial ones. As an example, the objects sizes (in number of packets) and the respective packets inter-arrivals have to be modeled carefully in order to reproduce accurately the original traffic spectrum. Nevertheless, our study demonstrated that the presence of heavy tailed distributions in traffic does not necessarily implies correlation, some of them can be modeled by memoryless distributions without impacting the traffic scaling properties.

In future works, we will investigate a new methodology to evaluate LiTGen's accuracy. We will compare the performance of a simple queue model fed, in the one hand by the captured traffic and, in the other hand by the synthetic traffic generated with LiTGen. The first results confirm LiTGen's ability to catch the captured traffic properties accurately: the performance of the queue under the synthetic traffic are very close to the ones obtained under the captured traffic, whereas simpler renewal processes (such as Poisson processes) as an input give performance parameters that are very far from reality. We will also investigate the dependency between object sizes and the packets arrival process as a possible signature for anomaly detection.

References

1. Mah, B.A.: An empirical model of http network traffic. In: IEEE Infocom, Kobe, Japan (April 1997)
2. Barford, P., Crovella, M.: Generating representative web workloads for network and server performance evaluation. In: ACM SIGMETRICS, Madison, Wisconsin, USA (June 1998)

3. Sommers, J., Barford, P.: Self-configuring network traffic generation. In: ACM IMC, Taormina, Sicily, Italy (October 2004)
4. Vishwanath, K.V., Vahdat, A.: Realistic and responsive network traffic generation. In: ACM SIGCOMM, Pisa, Italy (September 2006)
5. Crovella, M., Bestavros, A.: Self-similarity in world wide web traffic: Evidence and possible causes. In: ACM SIGMETRICS, Philadelphia, PA, USA (May 1996)
6. Willinger, W., Taqqu, M.S., Sherman, R., Wilson, D.V.: Self-Similarity throught high-variability: Statistical analysis of ethernet LAN traffic at the source level. In: ACM SIGCOMM, Philadelphia, PA, USA (August 1995)
7. Misra, V., Gong, W.B.: A hierarchical model for teletraffic. In: IEEE CDC, Tampa, Florida, USA (December 1998)
8. Ridoux, J., Nucci, A., Veitch, D.: Seeing the difference in IP traffic: Wireless versus wireline. In: IEEE Infocom, Barcelona, Spain (April 2006)
9. Rolland, C., Ridoux, J., Baynat, B.: LiTGen, a lightweight traffic generator: application to P2P and mail wireless traffic. In: PAM, Louvain-la-neuve, Belgium (April 2007)
10. Donelson-Smith, F., Hernandez-Campos, F., Jeffay, K., Ott, D.: What TCP/IP protocol headers can tell us about the web. In: ACM SIGMETRICS, Cambridge, Massachusetts, USA (June 2001)
11. : D. Veitch and P. Abry: Matlab code for the wavelet based analysis of scaling processes, http://www.cubinlab.ee.mu.oz.au/∼darryl/.
12. Abry, P., Taqqu, M.S., Flandrin, P., Veitch, D.: Wavelets for the analysis, estimation, and synthesis of scaling data. In: Self-Similar Network Traffic and Performance Evaluation. Wiley (2000)
13. Jiang, H., Dovrolis, C.: Why is the internet traffic bursty in short time scales? In: Sigmetrics, Banff, Alberta, Canada (June 2005)
14. Hohn, N., Veitch, D., Abry, P.: Does fractal scaling at the IP level depend on TCP flow arrival process? In: ACM IMC, Marseille, France (November 2002)
15. Rolland, C., Ridoux, J., Baynat, B.: Hierarchical models for different kinds of traffics on CDMA-1xRTT networks. Technical report, UPMC - Paris VI, LIP6/CNRS (2006) http://www-rp.lip6.fr/∼rolland/techreport.pdf.
16. Figueiredo, D., Liu, B., Misra, V., Towsley, D.: On the autocorrelation structure of TCP traffic. In: Computer Networks. Volume 40. (October 2002) 339–361