LiTGen, a lightweight traffic generator: application to P2P and mail wireless traffic

Chloé Rolland*, Julien Ridoux†, and Bruno Baynat*

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

Abstract. LiTGen is an easy to use and tune open-loop traffic generator that statistically models wireless traffic on a per user and application basis. We first show how to calibrate the underlying hierarchical model, from packet level capture originating in an ISP wireless network. Using wavelet and semi-experiments analysis, we then prove LiTGen's ability to reproduce accurately the captured traffic burstiness and internal properties over a wide range of timescales. In addition the flexibility of LiTGen enables us to investigate the sensitivity of the traffic structure with respect to the possible distributions of the random variables involved in the model. Finally this study helps understanding the traffic scaling behaviors and their corresponding internal structure.
Key words: traffic generator, scaling behaviors, energy plot, semi-experiments

1 Introduction

The limited resources of wireless access networks, the users' contracts diversity and mobility are particularities that greatly impact the design of traffic models. Traffic generators proposed in the past years modeled primarily web traffic. [1] and [2] proposed hierarchical models, but did not validate them against real traffic traces. Recently, [3] is an effort to generate representative traffic for multiple and independent applications. The model underlying this generator is not designed to specify the packet level dynamics neither to capture the traffic scaling structure. In [4], the authors argue that network characteristics must be emulated to reproduce the burstiness observed in captured traffic. Their traffic generator relies then on a third party, link and network layers emulator (requiring the use of 11 cutting-edge computers). Thus, this opaque emulator makes the investigation of the obtained traffic scaling structure more complex.

In this paper, we present LiTGen, a "**Li**ght **T**raffic **Gen**erator" that statistically models wireless traffic. LiTGen relies on a simple hierarchical description of traffic entities, most of them modeled by uncorrelated random variables and renewal processes. The confrontation of LiTGen to real traces captured on an operational wireless network[1] proves its ability to reproduce accurately, not only

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

the observed traffic scaling behaviors over a wide range of timescales, but also the intrinsic properties of the traffic. This design does not require to consider network or protocol characteristics (*e.g.* RTT, link capacities, TCP dynamics...) and allows fast computation executed on a commonplace computer. To the best of our knowledge, we are the first ones to produce synthetic wireless traffic that accurately reflects the first two orders of the packets arrivals time series.

In the rest of this paper, section 2 describes LiTGen, its underlying model and how it generates synthetic traces. Section 3 validates LiTGen's ability to reproduce the complexity of the original traffic correlation structure, for both mail and peer-to-peer (P2P) traffics. 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 generator

2.1 Underlying Model

Earlier works identified three possible causes of correlation in IP traffic: the presence of heavy-tailed distributions [5], the superimposition of independent traffic sources [6] and the inherent structure and interactions of protocol layers [7]. These two last assumptions call on the conception of our traffic generator to be based on a user-oriented approach and a hierarchical model. This model is made of several semantically meaningful levels, each of them characterized by a specific traffic entity. For each traffic entity, we define a set of random variables either related to a time or a size characterization.

Session level. We assume each user undergoes an infinite succession of session and inter-session periods. During a session, a user makes use of the network resources by downloading a certain number of objects. We define two random variables to characterize this level: $N_{session}$, the session size, *i.e.* the number of objects downloaded during a session and, T_{is}, the inter-session duration.

Object level. A session is made of one or several objects. Indeed, a session is split up into a set of requests (sent by user) and responses (from the server), where responses gather the session's objects. In the case of web, objects may be web pages' main bodies (HTML skeletons) or embedded pictures [8][2]. In the case of mail, objects may be servers responses to clients requests (*e.g.* e-mails, clients accounts meta-data...). In the case of P2P, objects may be files or chunks of files. The description of this level requires the definition of two random variables: N_{obj}, the object size, *i.e.* the number of IP packets in an object and, IA_{obj}, the objects inter-arrival times in a session.

Packet level. Finally, each object is made of a set of packets. The arrival process of packets in an object can be described by giving the successive inter-arrival times between packets, characterized by random variables IA_{pkt}.

[2] In this previous study applied to web traffic, the underlying model was made of four levels, including ***web pages level***. This extra level, not described here, is not relevant in the context of mail and P2P, but is kept for the generation of web traffic.

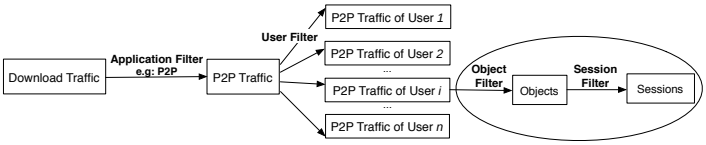


Fig. 1. Steps for filtering P2P traffic and identifying per-user traffic entities.

So far, we made no assumption concerning the random variables correlation structure. Indeed, inter-dependence mechanisms can be taken into account by introducing correlations between random variables. Of course, the objective here is to remain as simple as possible and to introduce dependencies only if necessary, as discussed in section 3. Note that one can equivalently remove from the hierarchy the session level by including the inter-session durations in the objects inter-arrivals distribution. Nevertheless, it would make the characterization of IA_{obj} more complex and LiTGen less easy to use in practice.

2.2 Wireless Trace Analysis

In order to calibrate and validate LiTGen underlying model, we benefit from data traces captured on the Sprint PCS CDMA-1xRTT access network. Traces have been captured on an OC-3 collecting link spanning a large geographical area and so tens of wireless access cells. The traffic capture consists in two unidirectional 24 hours long traces, captured simultaneously. Each of them is composed of a collection of IP packets with accurate timestamps and entire TCP/IP headers. Thus, we have access to the well-known 5-tuple: {IP destination, IP source, port destination, port source, transport protocol}. These traces have already been used in a previous study [9] that gives more details on the raw characteristics of this data traffic and its differences with wireline traffic.

Because of its small representation (less than 10%) in the traces and to narrow down the analysis, we exclude the UDP traffic from our study and focus on TCP traffic. Moreover, we are not interested in the modeling of the interactions between the upload and the download traffics. Indeed, we want to keep a very simple underlying model which do not rely on a network or TCP emulator. Finally, we focus on the traffic intended to the wireless terminals (download path). As a matter of fact, the upload wireless traffic contains mostly connection requests and ACKs, while the download wireless path is richer and has more importance from an operational point of view.

The model calibration requires to identify the characteristics of the traffic entities from the captured set of packets. To do so, we first filter traffic corresponding to a given application and then identify per-user traffic entities based on the 5-tuple associated to each packet (see Figure 1).

Packet Level. A filter based on a source port number selection retains traffic specific to a given application (*e.g.* 110, 143, 220 for the mail traffic). A user's packets share the same destination IP address and are then grouped into subsets of a given {IP source, port destination} pair, corresponding typically to the server IP address contacted and the port opened on the user's side. All resulting subsets

of a given user correspond then to all flows he requested (considering the given application).

Object Level. After applying the application and user filters, packets subsets are grouped to identify objects by means of the method presented in [10]. Based on the analysis of the TCP headers, this method observes the TCP flags (SYN, FIN, etc) to differentiate objects within packets subsets (characterized by a rupture in the acknowledgment number series).

Session Level. Finally, we aggregate objects into sessions. The definition of the sessions relies on *active periods* during which one or several objects are being downloaded. Those periods are separated by *inactive periods*. We use a temporal clustering method (also used in [1, 2, 10] for the retrieving of web pages) to infer the sessions' boundaries. An inactive period that lasts for more than a predefined threshold determines the precedent session termination. We fixed empirically the threshold to 300 seconds[3].

2.3 Traffic Generation

Contrarily to the trace analysis, LiTGen generates traffic from upper level entities (sessions) to lower ones (packets). LiTGen is used for the generation of traffic corresponding to different user's applications. For each kind of traffic, we first fix the number of users of the corresponding application. LiTGen generates then traffic for each user independently. The final synthetic trace is obtained by superimposing synthetic traffic of all users and all applications. For validation purposes we can extract the proportion and the number of users of each application from the captured trace. In an operational network these statistics can be derived from operator's knowledge of customer's subscribes services[4].

3 Validation

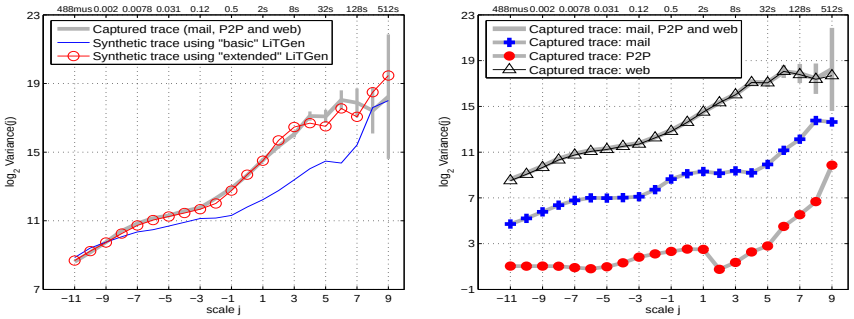
LiTGen is evaluated on its ability to reproduce the complexity of the traffic correlation structure in the captured packet traces. For this purpose, we use an energy spectrum comparison method to match the packets arrivals time series extracted from the original and corresponding synthetic traces. Since the 24-hour trace is not stationary, the analysis is performed on a one-hour period extracted from the entire trace. The results presented here correspond to a given one-hour period; similar results were obtained for other one-hour extracted traces.

3.1 Wavelet analysis

We use the Logscale Diagram Estimate or LDE [11] to perform analysis based on discrete wavelet transform. 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.

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

[4] Such a finite assumption is typically used for network planning to predict the active population that will be served during a given time.



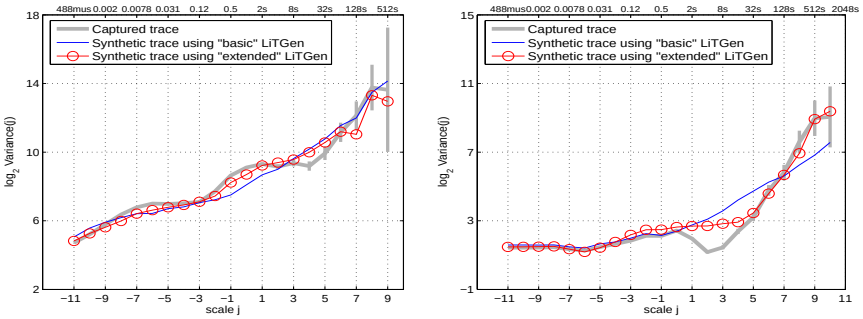
(a) "Basic" vs "extended" LiTGen (b) Mail, P2P and web spectra

Fig. 2. Model evaluation and comparison of the mail, P2P and web spectra.

We first focus on web, mail and P2P traffic and generate three independent synthetic traces using a simple version of our generator. With this so called *basic* LiTGen, all traffic entities are generated from renewal processes using the empirical distributions extracted from the captured trace. No other additional dependency is introduced between the random variables. The three synthetic traces are then merged into a single one and compared to the filtered captured traffic composed of the same three applications. Figure 2(a) shows the resulting LDE spectra. Clearly, the synthetic trace produced by basic LiTGen (thin curve) does not match the captured traffic spectrum (thick gray curve). This simple version of LiTGen's underlying model does not succeed in reproducing the captured traffic scaling structure with a good accuracy.

Previous studies (*e.g.* [9]) 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 model by introducing a dependency between the arrival process of packets within an object and the corresponding object's size. Note that this dependency may reflect the impact of TCP on packets inter-arrival times in objects of different sizes. In this extension, referred to as *extended* LiTGen, the arrival of packets within objects is still modeled by renewal processes, but for an object of a given size s, the inter-arrival random variables IA^s_{pkt} now depends on the object size. In order to evaluate extended LiTGen, we derive size-dependent empirical distributions of in-objects packets inter-arrivals, from the captured trace. When generating traffic, the packets inter-arrivals in an object of size s are taken from the corresponding IA^s_{pkt} distribution. The spectra obtained with extended LiTGen (circle curve in figure 2(a)) is barely distinguishable from the captured one. As a first result, the introduction of a simple dependency between the objects sizes and the packets inter-arrivals succeeds in reproducing accurately the traffic correlation structure, without taking into account network characteristics (such as TCP dynamics, RTT, loss rates). It thus appear that we do not need to introduce more complex non-renewal processes in the model, leading to a much simpler generator than the one developed in [4].

These three kinds of traffic, however, do not appear in the same proportions in the captured trace: while carrying 92.7% of the packets and 95.6% of the flows, web is the dominant application; mail carries 6.8% of the packets and



(a) Mail traffic (b) P2P traffic

Fig. 3. Model evaluation: "basic" VS "extended" LiTGen in mail and P2P traffic

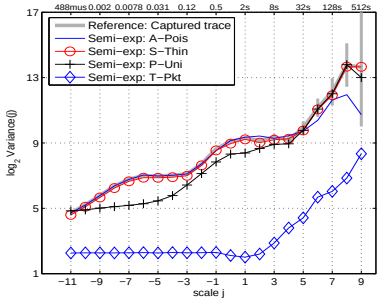
3.9% of the flows; P2P carries 0.5% of the packets and 0.5% of the flows. Figure 2(b) clearly indicates the differences between the three applications spectra in the captured trace that calls for studying each application independently. Figures 2(a) and 2(b) also show that our extended model accurately models the web traffic, conclusion reinforced by our previous study [8]. In the following, we thus focus on the mail and P2P traffics, which have been hidden by the predominant web traffic so far.

Figure 3(a) presents the mail traffic spectra. The reference spectrum (thick gray curve) corresponds to the captured mail traffic only. The basic LiTGen's underlying model (thin curve) reproduces in quite a good way the mail traffic spectrum. Extended LiTGen improves the results showing LiTGen's good ability to model mail traffic. Figure 3(b) shows the case of P2P traffic. Basic LiTGen fails to reproduce the captured traffic correlation structure for the scales above $j = 0$. The dip of reference spectrum at scales around $j = 2$ indicates a possible periodic behavior which we do not capture. Although the structural dependency introduced between N_{obj} and IA_{pkt} in extended LiTGen does not lead to the same improvement when dealing with P2P traffic, it allows the corresponding spectra to match the reference one (except over scales comprised between $j = 0$ and $j = 5$). Due to space limitation, we do not provide here the investigation required to capture this apparent periodic behavior and leave it as future work.

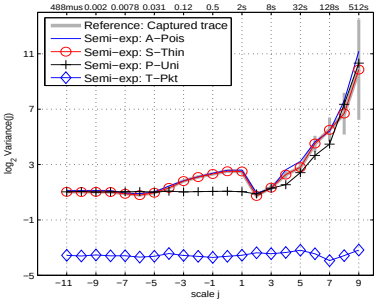
While LiTGen exhibits good results on the overall spectra, we need a further advanced methodology to validate the internal properties of the synthetic traffic.

3.2 Semi-experiments method

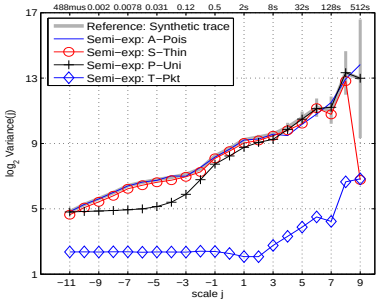
Semi-experiments have been introduced in [13] and consist in an arbitrary but insightful manipulation of internal parameters of the time series studied. The comparison of the energy spectrum before and after the semi-experiment leads to conclusions about the importance of the role played by the parameters modified by the semi-experiment. We apply the same set of semi-experiments to the captured traces and the synthetic traces generated by extended LiTGen. We then compare the impact of the internal manipulations to the two time series for the mail (Fig.4) and the P2P (Fig.5) traffic.



(a) Measured trace

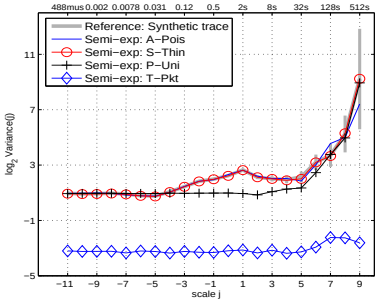


(a) Measured trace



(b) Synthetic (extended LiTGen)

Fig. 4. Semi-experiments: mail trace



(b) Synthetic (extended LiTGen)

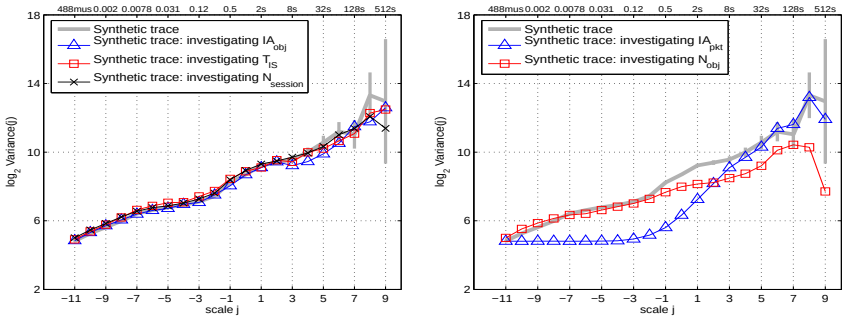
Fig. 5. Semi-experiments: P2P trace

T-Pkt is a **T**runcation manipulation that allows to examine the objects arrival process by keeping only the first packet of each object. Removing packets decreases the energy of the spectrum that takes smaller values. As shown in figures 4 and 5, **T-Pkt** has a similar impact on the captured and the synthetic traces, for both mail and P2P traffic.

The **S-Thin** manipulation allows to test 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 captured and synthetic trace, for the mail as well as P2P traffic, keep the same shape but drop by a small amount close to $log_2(0.9) = -0.15$.

A-Pois allows to examine the interactions between objects. This manipulation repositions the objects **A**rrival times according to a Poisson process and randomly permutes the objects order (while preserving the internal packet structure of objects). While **A-Pois** is a drastic manipulation, it has very little (and similar) effect on the spectra of all traces, indicating the negligible contribution of object arrival process in comparison to packets arrival.

P-Uni confirms this conclusion since it allows to examine the impact of in-objects packets burstiness. **P-Uni** 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 $j = -5$ for mail (resp. from scales $j = -11$ to $j = 2$ for P2P) in a comparable manner for the captured and synthetic traces.



(a) Insensitive: IA_{obj}, T_{IS} and $N_{session}$. (b) Sensitive: IA_{pkt} and N_{obj}

Fig. 6. Mail: test for the memoryless hypothesis

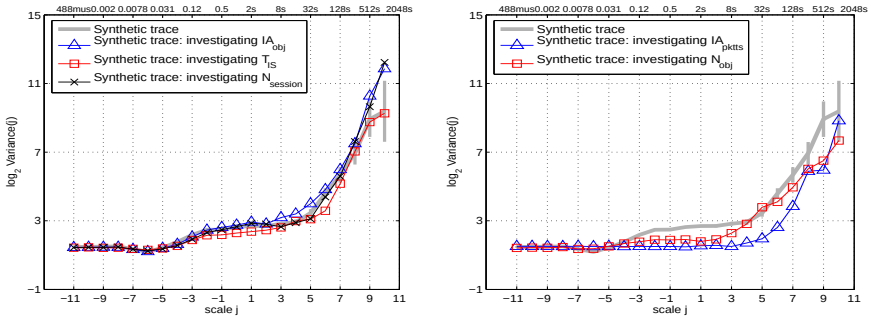
As a conclusion, the captured and synthetic traces spectra present similar reactions to each semi-experiment manipulation. This indicates that LiTGen captured the key internal properties of the traffic highlighted by the semi-experiments, *i.e.* the object arrival process has few influence on the traffic burstiness; the objects can be considered as independent and the packets arrival process within objects contributes mostly to the energy spectrum. Note that the simple structure of our traffic description, which still relies on renewal processes, is sufficient to reproduce these traffic internal properties.

4 Impact of traffic entities properties

We first investigate if "well-known" distributions can accurately approximate the empirical ones. To this aim, we use statistical quality of fit tests (*e.g.* KS-test) and compute indices of goodness of fit (*e.g.* Sum of Squares due to Error, R-Square...) to determine the "best" approximation. We led to similar conclusions for mail and P2P traffic. First, heavy-tailed distributions approximate well the random variables $N_{session}$ and N_{obj} (Pareto distributions) as well as IA_{obj} (Weibull distributions) and IA_{pkt} (close to lognormal distributions). Then exponential distributions approximate well T_{IS}.

The flexibility of extended LiTGen enables us to investigate the sensitivity of the traffic correlation structure with regards to the random variable distributions. To this aim, we replace individually the experimental distribution of each random variable by a memoryless distribution (exponential or geometric) of same mean. We thus create five synthetic traces, each one corresponding to a given random variable ($N_{session}$, T_{IS}, N_{obj}, IA_{obj} and IA_{pkt}). We then compare these traces to the reference synthetic trace generated by extended LiTGen calibrated with the experimental distributions.

Observing first the mail traffic, figure 6(a) shows that modeling the random variables IA_{obj}, T_{IS} and $N_{session}$ by memoryless distributions has a very small impact on the spectra of the LDE. On the contrary, modeling IA_{pkt} and N_{obj} by memoryless distributions widely impacts the spectra, as shown in figure 6(b). As an example, modeling IA_{pkt} by an exponential distribution completely erases the correlation existing between packets inter-arrivals and flattens the spectrum at



(a) Insensitive: IA_{obj}, T_{IS} and $Nsession$ (b) Sensitive: IA_{pkt} and N_{obj}

Fig. 7. P2P: test for the memoryless hypothesis

scales below $j = -3$. This confirms the results obtained by the semi-experiments methodology that designated the in-object packets inter-arrival structure as the main source of energy in the spectrum. As shown in figure 7 similar conclusions can be drawn for P2P traffic.

For both P2P and mail, we observe that modeling IA_{obj}, T_{is} and $N_{session}$ by memoryless distributions is a realistic assumption that leads to an extremely small loss of accuracy compared to the reference spectrum. On the contrary, modeling IA_{pkt} and N_{obj} by memoryless distributions is far from realistic and indicates the need to model these random variables more carefully.

This investigation leads then to very interesting results illustrated by the number of objects within sessions $N_{session}$ and their arrival process IA_{obj}. Although the experimental distributions of these random variables are closely approximated by heavy-tailed distributions, we show that both distributions have negligible influence on the scaling behaviors in traffic. The presence of heavy-tailed distributions does not compulsorily imply a presupposed scaling behavior.

Nevertheless, the internal structure of objects has a strong influence on the spectra. In both mail and P2P traffic, the investigation concerning the packets inter-arrival distribution clearly points it out as the source of correlation in traffic at small scales.

5 Conclusion

This paper describes LiTGen, a per-user oriented traffic generator. LiTGen has the benefit to reproduce accurately the traffic scaling properties at small and large time scales, while using a very simple underlying hierarchical model. Thanks to LiTGen, we investigated the impact of the random variables distributions describing the IP traffic structure. This investigation is important for two reasons. First it helps understanding the sensitivity of the traffic, with regards to the distributions involved in its description, and then identify crucial parameters. It also gives insights to anyone willing to provide accurate traffic models. Most analytical models rely on simple Markovian hypothesis and one must be careful about their impact. Whenever useful to improve accuracy, one should replace some of them (the proper ones) by more appropriate assumptions. As

an example, the exact original wireless traffic spectrum, can not be reproduced without 1) taking into account the organization of packets within objects 2) the use of heavy-tailed distributions to model objects size (in number of packets) and respective packet inter-arrivals. Moreover, our study demonstrated that the presence of heavy-tailed distributions in traffic does not necessarily implies the correlation, some of them can be modeled by memoryless distributions without impacting the traffic scaling properties.

This study also demonstrated the ability of a hierarchical model to reproduce accurately the characteristics of classes of traffic. The exhibition of results corresponding to other datasets is part of our ongoing works. Precise classes of applications (*e.g.* web and mail together, the main contributors to wireless traffic) will be defined soon to specify the utilization domain of the hierarchical model. While the results of LiTGen are proper for the P2P traffic, a more accurate but simple model for P2P traffic and other classes of application is still to be defined. This will become particularly important when mobile users will massively adopt new services and decrease the domination of web application in the overall traffic.

References

1. Mah, B.A.: An empirical model of http network traffic. In: IEEE Infocom. (1997)
2. Barford, P., Crovella, M.: Generating representative web workloads for network and server performance evaluation. In: ACM Sigmetrics. (1998)
3. Sommers, J., Barford, P.: Self-configuring network traffic generation. In: ACM IMC. (2004)
4. Vishwanath, K.V., Vahdat, A.: Realistic and responsive network traffic generation. In: ACM Sigcomm. (2006)
5. Crovella, M., Bestavros, A.: Self-similarity in world wide web traffic: Evidence and possible causes. In: ACM Sigmetrics. (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. (1995)
7. Misra, V., Gong, W.B.: A hierarchical model for teletraffic. In: IEEE CDC. (1998)
8. 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.
9. Ridoux, J., Nucci, A., Veitch, D.: Seeing the difference in IP traffic: Wireless versus wireline. In: IEEE Infocom. (2006)
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. (2001)
11. Veitch, D. and Abry, P.: 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. Hohn, N., Veitch, D., Abry, P.: Does fractal scaling at the IP level depend on TCP flow arrival process? In: ACM IMC. (2002)