Using LiTGen, a realistic IP traffic model, to evaluate the impact of burstiness on performance

Chloé Rolland
Université Pierre et Marie Curie – Paris VI
LIP6/CNRS, UMR 7606
Paris, France
Chloe.Rolland@lip6.fr

Julien Ridoux
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

Bruno Baynat
Université Pierre et Marie Curie – Paris VI
LIP6/CNRS, UMR 7606
Paris, France
Bruno.Baynat@lip6.fr

Vincent Borrel
Université Pierre et Marie Curie – Paris VI
LIP6/CNRS, UMR 7606
Paris, France
Vincent.Borrel@lip6.fr

ABSTRACT

For practical reasons, network simulators have to be designed on traffic models as realistic as possible. This paper presents the evaluation of LiTGen, a realistic IP traffic model, for the generation of IP traffic with accurate time scale properties and performance. We confront LiTGen against real data traces[1] using two methods of evaluation. These methods respectively allow to observe the causes and consequences of the traffic burstiness. Using a wavelet spectrum analysis, we first highlight the intrinsic characteristics of the traffic and show LiTGen's ability to reproduce accurately the captured traffic correlation structures over a wide range of timescales. Then, a performance analysis based on simulations quantifies the impact of these characteristics on a simple queuing system, and demonstrates LiTGen's ability to generate synthetic traffic leading to realistic performance. Finally, we conduct an investigation for a possible model reduction using memoryless assumptions.

Categories and Subject Descriptors

C.2.m [**Computer-Communication Networks**]: Miscellaneous; C.4 [**Performance of Systems**]: Measurement techniques, Modeling techniques, Performance attributes

General Terms

Measurement, Performance.

Keywords

Traffic generator, scaling behaviors, second-order analysis, performance evaluation.

1. INTRODUCTION

In the past years, numerous studies highlighted evidences for scaling behaviors in IP traffic. Local-area traffic and web traffic have been shown to be self-similar [1, 2]; long-range dependence [3] and fractal behaviors [4] have been discovered in backbone traffic. These scaling properties indicate the correlations existing in the time series extracted from the traffic traces. The correlation structure of the traffic has strong implications on queuing and performance, especially leading to high variability (burstiness) over a wide range of time scales [5].

To be relevant from a practical point of view, network simulators have to be designed on traffic models as realistic as possible. A wide range of proposed models exist, usually sacrificing accuracy in favour of simplicity. A simple but inaccurate Poisson or renewal process offers a solution to packet level modeling. More accurate examples are complex stochastic processes (*e.g.* Fractional Gaussian Noise, Fractional ARIMA processes, Multifractal Wavelet Model) [1, 6, 7]. Additionally, hybrid solutions propose the use of emulators to reproduce the characteristics of lower layers of the protocol stack [8]. All these models aim at reenacting the observed traffic correlation structures.

The selection of a model of traffic is then a trade-off between accuracy and complexity and depends on practical needs and constraints. In [9, 10] we revisit a simple hierarchical model of traffic that has been shown to reproduce fairly accurately the traffic correlation structures. Thanks to a combination of wavelet spectrum and semi-experiments analysis of traffic traces, we show the ability of the associated traffic generator (LiTGen) to produce traces that exhibit correlation structures similar to real traffic traces.

In this paper, we strengthen and refine the results relative to the spectrum analysis thanks to a complementary method of validation. Using a queuing model, we simulate the outcome of the synthetic traffic produced by LiTGen. LiTGen's ability to produce realistic traffic traces is evaluated by comparing the performance parameters of a simple queue fed by real traffic traces and synthetic traces generated with LiTGen under simulation. By combining both evaluation methods, we are then able to discuss the causes and the consequences of time-scale behaviors in IP traffic, and quantify the gap between the model and the reality. We finally observe the sensitivity of the system with regards to the distributions involved in the generated traffic, by replacing each of them with memoryless

[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.

SIMUTOOLS 2008, March 03-07, Marseille, France
Copyright © 2008 ICST 978-963-9799-20-2
DOI 10.4108/ICST.SIMUTOOLS2008.3065

distribution and evaluate the impact on performance.

The following section 2 presents the hierarchical model of traffic and the corresponding traffic generator LiTGen. Section 3 provides an evaluation of LiTGen and highlights the similarities between the two evaluation methods. Finally section 4 explores the impact of the model reduction.

2. LITGEN, A LIGHTWEIGHT TRAFFIC GENERATOR

2.1 Underlying Model

In our previous works [9, 10], we presented LiTGen, a traffic generator, based on (i) an application-oriented approach, (ii) a user-oriented approach and (iii) a semantically meaningful hierarchical model, which depends on the studied application. This model is made of several levels, each of them characterized by a specific traffic entity. Taking the example of web traffic, we consider the specific model presented in [10]. In this model, we assume each user undergoes an infinite succession of **session** and inter-session periods. During a session, a user downloads a certain number of web **pages**, separated by reading times (OFF periods). 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 files or embedded objects such as pictures). Finally, each object is made of a set of **packets**.

Each network entity defined is represented by one or several random variables either related to a duration or a size metric. $N_{session}$ describes the user's session size, counting the number of pages downloaded, while T_{IS} characterizes the inter-session duration. The page size N_{page} indicates the number of objects composing a page and T_{off} the corresponding reading duration. IA_{obj}, N_{obj} and IA_{pkt} respectively characterize the objects inter-arrival in a page, the number of packets and their inter-arrivals in an object.

The model structure is related to the studied application. The page level is obviously specific to the web traffic and does not make sense for most of the other kinds of traffic. Sessions, objects (*e.g.* mail servers responses, chunks of files from peers in Peer-To-Peer traffic) and packets are the only entities required to model most applications and are common to all of them [9].

2.2 Calibration

To calibrate LiTGen, 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, providing then a 5-tuple per packet captured.

We focus here on the traffic intended to the users terminals (download path). As a matter of fact, the upload traffic contains mostly connection requests and ACKs, while the download path is richer and has more importance from an operational point of view. Moreover, we do not model the client/server interactions in upload and download directions. Also the underlying model does not rely on a network or TCP emulator that would reproduce the link layer or TCP dynamics [9, 10]. Because of its small representation (less than 10%) in the traces, we exclude the UDP traffic from our study and focus on TCP traffic [11, 12, 13]. Since the 24-hour trace is not stationary [14], the analysis is performed on a one-hour long period extracted from the entire captured trace. The results presented in this paper correspond to a given one-hour period; similar results were obtained for the other one-hour extracted traces.

The model calibration consists in providing a distribution to each of the random variables defined in the model. We first rely on the captured trace for this purpose, requiring to identify the traffic entities from it. This aggregation is based on the 5-tuple associated to each packet {IP destination, IP source, port destination, port source, transport protocol}. A filter based on a source port number selection retains packets specific to a given application (*e.g.* {80, 8080, 443} for the web application). A user's 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 flows the user requested, considering the given application.

Using techniques developed in [9, 10], we identify objects within the packets sets. In the case of web traffic (resp. mail and P2P), we then aggregate objects into web pages and finally aggregate web pages into sessions (resp. objects into sessions), relying on heuristics using temporal thresholds.

LiTGen is used for the generation of traffic corresponding to different user's applications. 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[2]. LiTGen then generates traffic for each user independently, from upper level entities (sessions) to lower ones (packets). The final synthetic trace is obtained by superimposing synthetic traffic of all users and all applications. More details about the generation process can be found in [10].

2.3 Basic and extended LiTGen

In our previous works [9, 10], we presented two versions of our generator LiTGen: *basic* LiTGen and *extended* LiTGen. In basic LiTGen, all traffic entities are generated from renewal processes, and no dependency of any kind is introduced between the random variables. We showed that basic LitGen did not succeed in reproducing the captured traffic burstiness with a good accuracy, requiring to refine it. To remain as simple as possible, and in order to avoid the need of a network or TCP emulator, we refined LiTGen by only introducing a dependency between N_{obj} and IA_{pkt}. In this so called extended LiTGen, the in-objects packets arrivals are still modeled by renewal processes but, the average in-object packets inter-arrival times now depend on the corresponding object size. We showed in [9, 10] that extended LiTGen significantly improves the results obtained with basic LiTGen.

3. LITGEN'S EVALUATION

We use LiTGen to illustrate the capacity of the hierarchical model to capture the complexity of the traffic correlation structure and to generate synthetic traffic leading to realistic performance. To this aim, we use two complementary methods. On the one hand, an energy spectrum comparison method allows us to match the packets arrivals time series extracted from the captured trace and the corresponding synthetic trace produced by LiTGen. On the other hand, a set of queuing system simulations allows us to evaluate LiTGen's ability to predict realistic performance.

[2] In an operational network these statistics can be derived from operator's knowledge of customers subscribed services.

3.1 Methods of validation used

3.1.1 Wavelet based analysis

We first use the Logscale Diagram Estimate or LDE [15] to perform a discrete wavelet transform analysis. This method gives insights on the intrinsic characteristics of a traffic trace seen as a time series of packets arrivals. For such a given time series, the LDE produces a logarithm plot of the data wavelet spectrum estimates. These plots are particularly useful to observe the variation of energy at different octaves or time scales. While straight lines with positive slope constitute experimental evidence for the presence of scaling, an horizontal alignment of points indicates the absence of scaling in the traffic (*e.g.* Poisson process). More details about mathematical developments and intuitions behind this technique can be found in [16]. This tool allows an energy spectra comparison that we use to assess the accuracy of the synthetic traces produced by LiTGen.

3.1.2 Performance analysis

We then study the impact of the synthetic traffic produced by LiTGen on a simple queuing model [5]. This queue is a simple representation of any network element whose performance are worth being observed. A more complex model (such as a queuing network) could have been studied if it was the result of a constructive modeling of the system, but would probably have given similar conclusions, as in most queuing systems the bottleneck queue drives the performance of the whole network.

For each simulation run, we use a queuing system with an infinite waiting room and a single server. From these simulations, we observe the waiting time and the queue size performance metrics. Assuming the average packets inter-arrival time in the system is $1/\lambda$, we simulate the queue behavior with different average service times $1/\mu$ insuring stability (*i.e.* such that $\lambda < \mu$), taken from a deterministic ($cv^2 = 0$), exponential ($cv^2 = 1$) or Coxian-2 (with $cv^2 = 5$) distribution.

3.2 Results

For each kind of application traffic, we consider five traffic traces: the captured trace; a synthetic trace generated with extended LiTGen; a synthetic trace generated with basic LiTGen; a synthetic trace generated from a single renewal process (SRP), modeling the packets arrivals in the system with the empirical distribution (without considering any underlying model, nor notion of flow, nor user point of view); and a synthetic trace generated from a Poisson process.

We first focus on web traffic. Among the three kinds of applications we study, the web application is the dominant one. While web traffic carries 92.7% of the packets and 95.6% of the flows, mail carries 6.8% of the packets and 3.9% of the flows, and P2P carries 0.5% of the packets and 0.5% of the flows.

Figure 1 presents the spectra resulting from the wavelet based analysis. We first observe that modeling the packets arrivals using a single Poisson process leads (non surprisingly) to an horizontal alignment of points (diamond line), indicating the absence of scaling. A single renewal process (star line) modeled with the empirical packets inter-arrivals distribution, also fails in capturing the traffic burstiness. This indicates that reproducing only the packets inter-arrivals distribution, without taking into account the correlation existing between successive inter-arrivals, is not sufficient to catch the traffic scaling properties. Figure 1 highlights the better results obtained with basic LiTGen (thin curve) and the importance of the underlying traffic structure on the bustiness. Still, the spectrum of the synthetic trace produced by basic LiTGen is far different from the captured trace spectrum. Their spectra differ from the octave $j = -9$, which is also the point of deviation of the spectra of the captured trace and synthetic trace obtained from a single renewal process. The corresponding observation timescale, 2 milliseconds, is in the order of the average inter-arrival time of packets within objects, pointing out that a great part of energy is due to the organization of packets within objects. Finally, the spectrum corresponding to the synthetic trace obtained with extended LiTGen (circle line) is barely distinguishable from the captured one, showing extended LiTGen's good ability to reproduce accurately the traffic correlation structure [10].

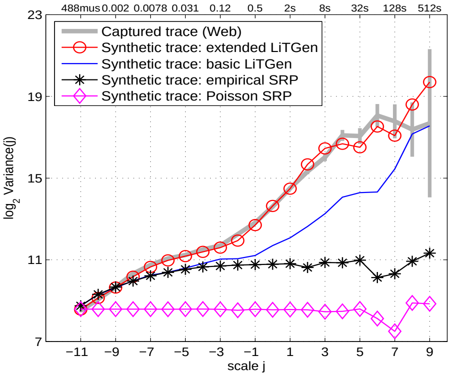
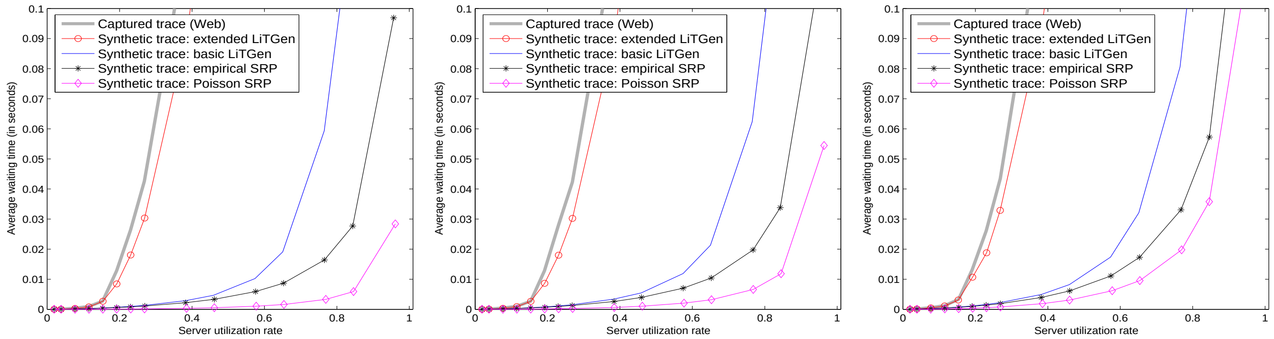


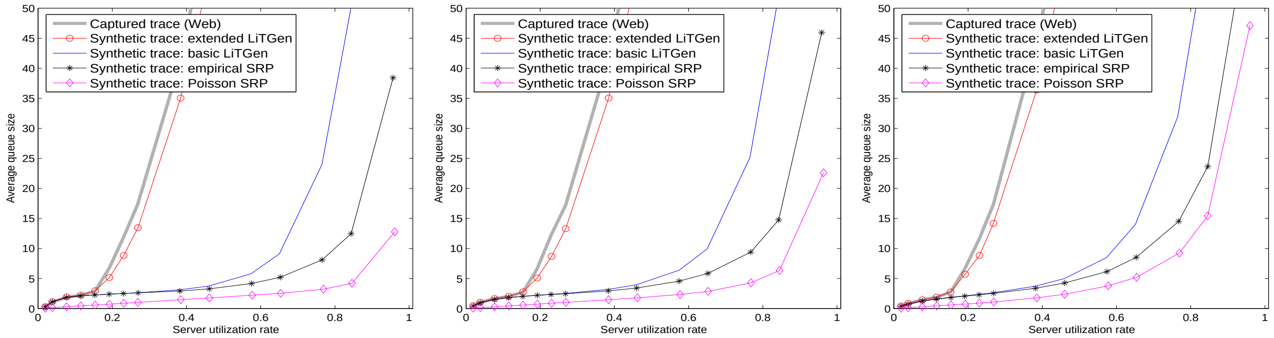
Figure 1: Wavelet analysis - Web traffic

We now observe the impact of these different traffic traces properties on a system made of a simple queue. We feed the three queues ($cv^2 = 0; 1; 5$) presented in section 3.1.2 with the five traffic traces. We first plot the average waiting time (figure 2) and the average queue size (figure 3) against the server utilization rate, for each average service time $1/\mu$ considered and for each input trace we study. Figure 2(b) presents the average waiting time for a G/M/1 queue system. First, we observe that the curves corresponding to the captured trace and to the synthetic trace obtained with extended LiTGen are really close, but not superimposed. Extended LiTGen performs much better than basic LiTGen. Although basic LiTGen exhibits long range dependency, the corresponding synthetic traffic leads to exaggerated optimistic performance, much closer to the curves corresponding to the use of a single renewal process than the one corresponding to extended LiTGen. Basic LiTGen however still gives better results than the use of renewal processes (based on empirical distributions or Poisson distributions) that aggregates blindly all the packets arrivals and predict over-optimistic performance. As a first conclusion, the performance analysis provides consistent qualitative results with the spectrum analysis, in the sense that extended LiTGen gives better performance than Basic LiTGen, which in turn gives better performance than any renewal process. However the performance analysis reveals unexpected deviations on the average performance parameters between the different traces, that would suggest the need for a more precise modeling of the in-object packets arrival process. Note that we obtained very similar results when modeling the service time by a deterministic law, an exponential law, or a Coxian-2 law with a large cv^2. In practice, the average waiting time and queue size increase with the cv^2 of the law considered but the qualitative behaviors and differences between the different traces are similar. For conciseness we show in the rest of this paper results corresponding to G/M/1 queues.



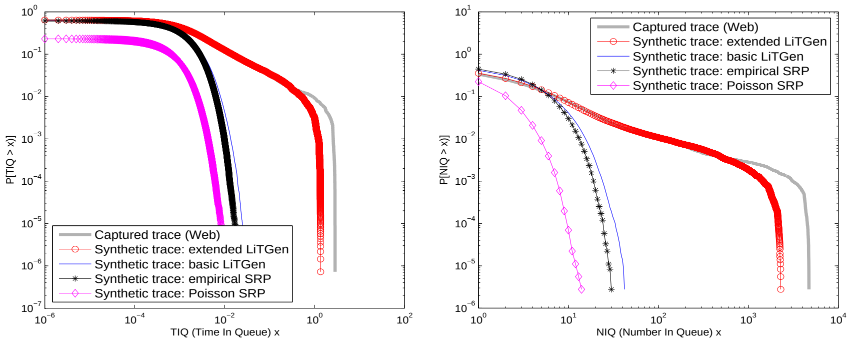
(a) G/D/1 (b) G/M/1 (c) G/Cox-2/1

Figure 2: Average waiting time vs. server utilization rate - Web traffic



(a) G/D/1 (b) G/M/1 (c) G/Cox-2/1

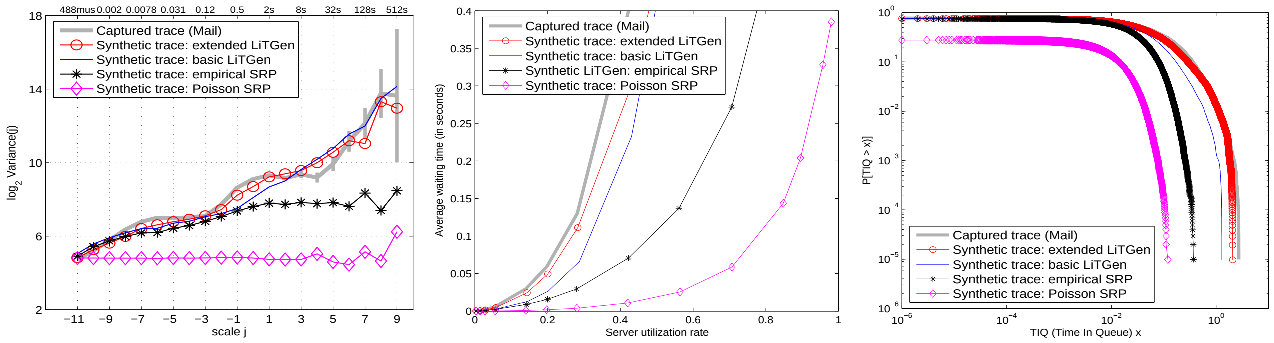
Figure 3: Average queue size vs. server utilization rate - Web traffic



(a) Waiting time distribution (b) Size of the queue distribution

Figure 4: Performance parameters distributions (CCDF) for the Web traffic, G/M/1, $\rho = 0.23$

We now consider the waiting time and queue size distributions. The distributions are obtained at a server utilization rate of about $\rho = 0.23$, located on the rise area of the average curve, which is of particular interest from a traffic engineering point of view [5]. Note that we draw the distributions corresponding to greater server utilization rates and we obtained similar results. Figure 4(a) (resp. 4(b)) presents the complementary cumulative distribution functions (CCDF) of the waiting time (resp. queue size) in a Log-Log scale for each input trace. In figure 4(a), we first observe that the CCDF obtained from the captured trace and the one obtained from extended LiTGen are almost superimposed for the first 98 percentiles of the distributions, and diverge for waiting times greater than 0,5 seconds. Even if this occurs with a very small probability (less than 10^{-2}), the real trace results in large waiting times that cannot be reached with the generated traces. Similar and even more trenchant comments stand for the queue size distribution (see figure 4(b)). These results confirm the need for a refinement of the object internal structure, in order to respect with a better accuracy the distribution of the performance parameters. The other three synthetic traces differ largely from the captured one. In particular, we observe that modeling the input as a renewal process using the empirical distribution widely underestimates queuing delays and queue size. In this last case, the maximum queue size is 33 packets (resp. the maximum waiting time is 0.02 seconds), while the maximum queue size is 4722 (resp. the maximum waiting time is 2.84 seconds) for the captured trace. Similarly, while the curve corresponding to basic LiTGen fits the captured one for a large number of values, it still underestimates the performance metrics. Indeed, considering the waiting time distribution (resp. the size of the queue distribution), they differ from the captured trace near $x = 4$ packet (resp. near $x = 1$ milliseconds).

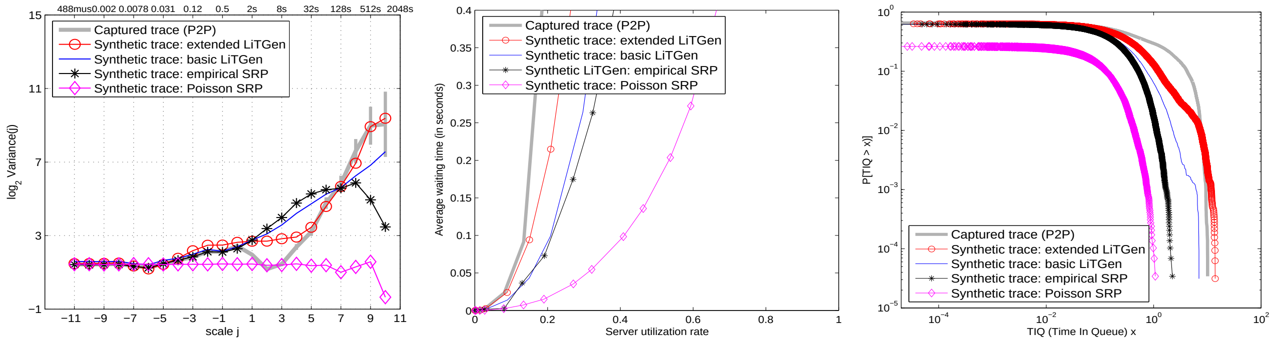


(a) Wavelet analysis

(b) Average waiting time vs. server utilization rate

(c) Waiting time distribution

Figure 6: Mail traffic



(a) Wavelet analysis

(b) Average waiting time vs. server utilization rate

(c) Waiting time distribution

Figure 7: P2P traffic

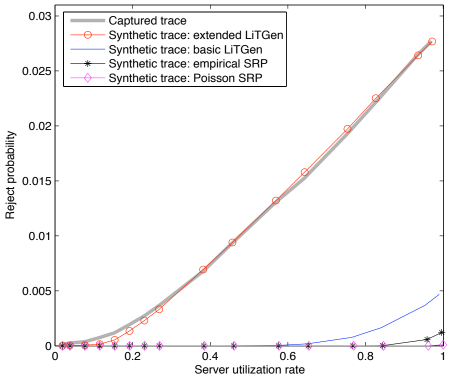
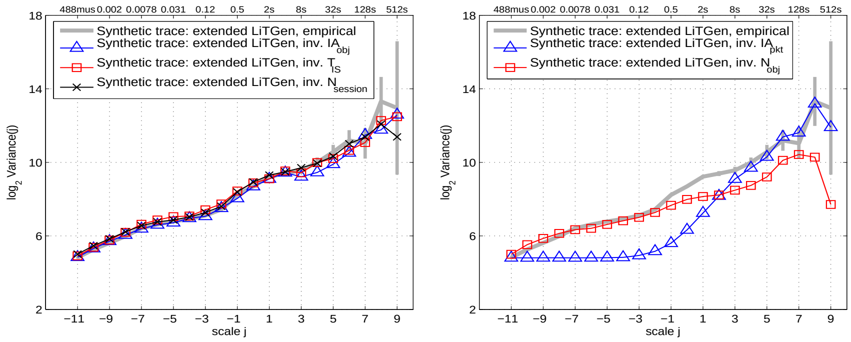


Figure 5: Reject Probability vs. server utilization rate - Web traffic, G/D/1/100

Finally, figure 5 presents the reject probability against the server utilization rate, for a G/D/1/100 system (queue limited to 100 packets). As can be seen in the figure, the curve corresponding to extended LiTGen and the one corresponding to the captured trace are almost superimposing, leading to high reject probabilities, whatever traffic load considered. Thus, we manage to reproduce in a very good way the traffic burstiness. At the opposite, we observe that Poisson process (as well as basic LiTGen) leads to a very bad estimation of the reject probability, even for light traffic loads.

We now quickly focus on mail traffic (see figure 6) and P2P traffic (see figure 7). We basically observe similar results in comparison to web traffic. In the case of mail traffic, the difference between the spectra corresponding to basic and extended LiTGen is however not as obvious as in the case of web traffic, as seen in figure 6(a). Figures 6(b) and 6(c) confirm this observation. Moreover, the refinement introduced in extended LiTGen does not lead to the same improvement when dealing with P2P traffic (see figures 7(a), 7(b) and 7(c)). Finally, the spectrum corresponding to the trace obtained with basic LiTGen (thin line) and the one corresponding to the trace from the single empirical renewal process used (star line) are really close.

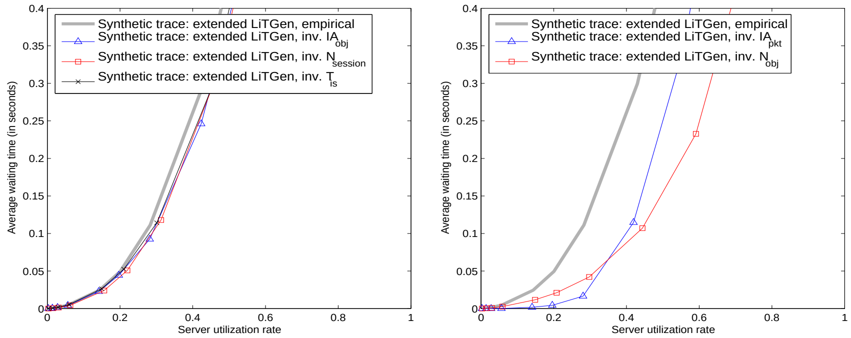
In conclusion, with regards to the qualitative aspect, we obtain very consistent results when investigating the traffic scaling structure and the consequences of this structure on the performance of a simple queuing system. The traces that exhibit more accurate scaling behaviors lead to more realistic performance previsions. In the other hand, considering the quantitative aspect, the performance analysis provided in this paper reveals stronger differences between the different traces, than those observed with the wavelet based study. In addition, these deviations are easier to quantify and interpret, as they deal with more tangible parameters than energy. Finally, this study emphasizes the need for a finer modeling of the in-objects dynamics, e.g. by introducing an embedded “Burst” level in the packet arrival process. This will be confirmed in the following section.



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

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

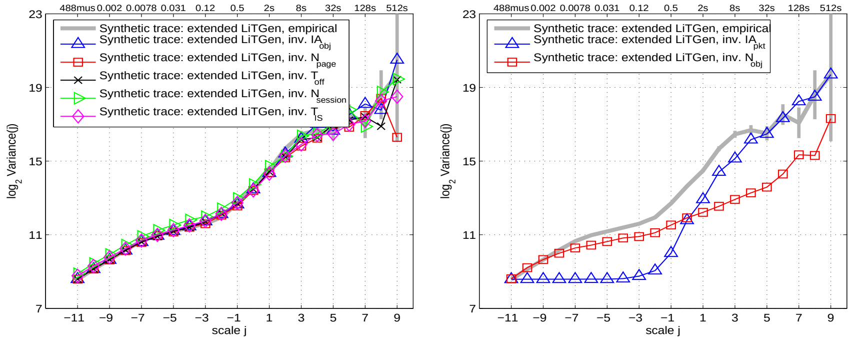
Figure 8: Mail: testing random variables memoryless hypothesis - wavelet analysis



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

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

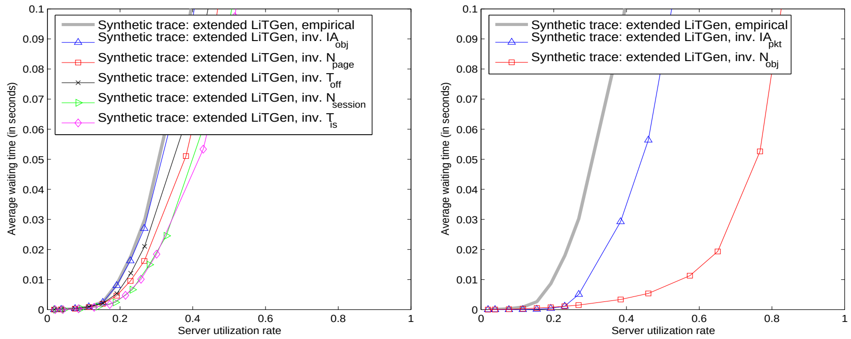
Figure 9: Mail: testing random variables memoryless hypothesis - performance analysis



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

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

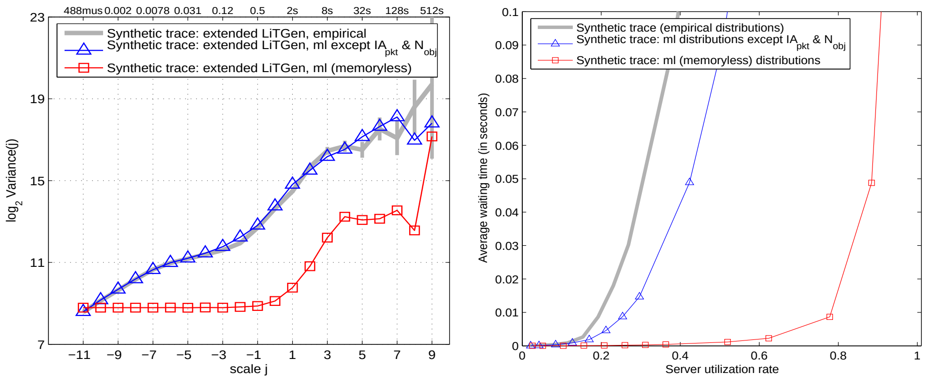
Figure 10: Web: testing random variables memoryless hypothesis - wavelet analysis



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

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

Figure 11: Web: testing random variables memoryless hypothesis - performance analysis



(a) Web traffic

(b) Web traffic

Figure 12: Investigating the substitution of all the empirical distributions

4. SENSITIVITY OF THE TRAFFIC WITH REGARD TO THE DISTRIBUTIONS

While several works studied the relationship between burstiness, and network or protocol characteristics (*e.g.* loss probabilities, RTT, link capacities, TCP dynamics) [11, 12, 17], LiTGen's flexibility allows us to investigate the impact of random variables potentially contributing to the traffic burstiness. To this aim, we individually replace the empirical distribution of each random variable by a memoryless distribution (exponential or geometric) of same mean. We thus create several synthetic traces (seven in the case of web traffic and five in the case of mail or P2P traffic), each one corresponding to a given random variable being replaced. We then compare the corresponding traces to the reference synthetic trace generated by extended LiTGen calibrated with the empirical distributions. Based on the wavelet spectra and the performance results, this comparison distinguishes the random variables whose modeling with a memoryless assumption have a significant impact, from those which have a negligible impact.

Observing first the mail traffic, figure 8 illustrates the sensitivity of the traffic burstiness against the distributions of the random variables of the underlying model. Modeling the distributions of the random variables IA_{obj}, $N_{session}$ and T_{IS} by memoryless distributions leads to very few impact on the data spectra, as shown in figure 8(a) in which the synthetic traces are barely distinguishable from the reference one. The traffic scaling structure of the studied trace seems to be quite insensitive with respect to the distributions of these random variables. On the contrary, we see in figure 8(b) that modeling the random variables IA_{pkt} and N_{obj} by memoryless distributions widely impacts the spectra. While modeling IA_{pkt} by an exponential distribution flattens the spectrum at scales below $j = -3$, modeling N_{obj} by a geometric distribution removes energy at large scales. The traffic burstiness seems then very sensitive with respect to the distributions of these two random variables.

Figure 9 presents the corresponding performance analysis of the same types of traces still for mail traffic: we again plot the average waiting time *vs.* the server utilization rate. Looking at figure 9(a), which shows the performance analysis of the random variables highlighted as *insensitive* in the previous paragraph, we observe that, although the curves are not superimposing, they are close to the reference one. Figure 9(b) presents the results concerning the *sensitive* random variables and show their over-optimistic impact on the performance.

Very similar results have been found for P2P traffic and are omitted here for clarity. Following the same steps, the investigation leads to the same two groups of random variables and the results obtained thanks to the wavelet and the performance analysis remain consistent.

Figure 10 presents the wavelet based analysis for web traffic. As shown on figure 10(a), the random variables IA_{obj}, N_{page}, T_{off}, $N_{session}$ and T_{IS} modeled by memoryless distributions lead to negligible impact on the data spectra. As seen in the case of mail and P2P traffic, the random variables IA_{pkt} and N_{obj} also have a great impact on the spectra (see figure 10(b)), highlighting the sensitivity of the traffic with regards to the distributions of these random variables. Compared with mail traffic, however, we observe stronger differences between the wavelet based analysis and the performance analysis. Figure 11(a) shows that the investigation of some of the random variables highlighted as insensitive during the wavelet analysis leads to less accurate performance prevision. This is notably the case of T_{IS}, $N_{session}$ and N_{page}. The modification of the time series for these three random variables impact more widely the final performance and challenge the apparent good results obtained with the wavelet analysis. This observation shows the importance of the two complementary validation methods we use here, and highlights again the need for a fine modeling of the in-object structure.

We finally create two synthetic traces, which we compare to the reference one. We obtain the first trace by calibrating the insensitive random variables with memoryless distributions. We thus create a synthetic trace in which only N_{obj} and IA_{pkt} are calibrated with empirical distributions. We present the results corresponding to the web traffic in figure 12, for the wavelet based analysis and the performance analysis. The results of the wavelet analysis (see figure 12(a)) show that the corresponding spectrum (triangle line) matches the reference one. However, a clear difference appears when looking at the waiting time performance metric between the investigated synthetic trace and the reference one, in figure 12(b). Even if the corresponding performance is more realistic than in the case of basic LiTGen or SRP, the difference between the two validation methods appears clearly in this case. The last synthetic trace is obtained by modeling *all* the random variables with memoryless distributions. In figure 12(a), we observe a great deviation between the corresponding spectrum (square line) and the reference one. In figure 12(b), the corresponding curve differs from the two others in a very strong way: the predicted performance are excessively optimistic. This confirms the importance for a very accurate modeling of IA_{pkt} and N_{obj}, in order to catch the captured traffic behavior and predict realistic performance. Note that we observed similar behaviors for the mail and P2P traffics.

As a conclusion of this section, the performance analysis gives finer insights about the sensitivity of the random variables involved in the underlying LiTGen model. Even if these random variables can be classified into two groups, some of the variables that have been declared as "insensitive" by the wavelet analysis have actually a non-negligible effect on the performance. There are two possible (and compatible) conclusions to this observation. First, the corresponding random variables cannot be modeled very accurately by memoryless distributions. Second, one must think of improving the underlying model, which again, drives us to the idea of a finer modeling of in-objects packets arrival process.

5. CONCLUSION

This paper describes the evaluation of LiTGen, a lightweight traffic generator. Illustrated on several types of traffics (web, mail and P2P), we confront the synthetic traces produced by LiTGen to two methodologies of evaluation. Allowing us to observe the causes and the consequences of correlation structures in the IP traffic, these two methodologies are complementary. While the wavelet spectrum analysis enables to highlight the traffic correlation structures over a wide range of timescales, the performance analysis quantifies the impact of these correlations on a simple queue modeling any critical bottleneck element in the network.

On the contrary to simpler models, LiTGen introduces a structural dependency between the average packets inter-arrival times and the objects sizes that succeeds in reproducing the captured traffic burstiness with a good accuracy. We show in this paper that the corresponding synthetic trace produced by LiTGen lead to realistic performance parameters.

We then investigate, thanks to LiTGen, the importance of a precise modeling of the distributions of the random variables involved in the underlying model. We could classify these random variables into "sensitive" and "insensitive" groups. This investigation emphasizes the contribution of the performance analysis, by bringing to light unexpected differences that did not appear with the wavelet based analysis. For example, some random variables actually have a mild but non negligible impact on the performance, while they appear to be "insensitive" if observed with the wavelet analysis only.

Finally, the study conducted in this paper underlined that the internal structure of embedded objects is crucial and has to be modeled very carefully to produce realistic traffic. In future works we will focus on the in-object structure and investigate a finer modeling of this structure using packets bursts, always with the objective of finding a compromise between simplicity (of the underlying model) and accuracy (of the resulted performance).

6. REFERENCES

[1] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of Ethernet traffic. In *ACM SIGCOMM*, 1993.

[2] M. Crovella and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. In *ACM SIGMETRICS*, May 1996.

[3] N. Hohn, D. Veitch, and P. Abry. Does fractal scaling at the IP level depend on TCP flow arrival process? In *ACM IMC*, November 2002.

[4] Z.-L. Zhang, V. J. Ribeiro, S. B. Moon, and C. Diot. Smalltime scaling behaviors of Internet backbone traffic: an empirical study. In *IEEE INFOCOM*, 2003.

[5] A. Erramilli, O. Narayan, and W. Willinger. Experimental Queuing Analysis with Long-Range Dependent Packet Traffic. In *IEEE/ACM Transactions on Networking*, 1996.

[6] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-Similarity throught high-variability: Statistical analysis of ethernet LAN traffic at the source level. In *ACM SIGCOMM*, August 1995.

[7] R. H. Riedi, M. S. Crouse, V. J. Ribeiro, and R. G. Baraniuk. A Multifractal Wavelet Model with Application to Network Traffic. *IEEE Transactions on Information Theory*, 45(4):992–1018, 1999.

[8] J. Sommers and P. Barford. Self-configuring network traffic generation. In *ACM IMC*, October 2004.

[9] C. Rolland, J. Ridoux, and B. Baynat. LiTGen, a lightweight traffic generator: application to P2P and mail wireless traffic. In *PAM*, April 2007.

[10] C. Rolland, J. Ridoux, and B. Baynat. Catching IP traffic burstiness with a lightweight generator. In *IFIP Networking*, May 2007.

[11] H. Jiang and C. Dovrolis. Why is the internet traffic bursty in short time scales? In *Sigmetrics*, June 2005.

[12] D.R Figueiredo, B. Liu, V. Misra, and D. Towsley. On the autocorrelation structure of TCP traffic. In *Computer Networks*, volume 40, pages 339–361, October 2002.

[13] A. Feldmann, A.C Gilbert, P. Huang, and W. Willinger. Dynamics of IP traffic: A study of the role of variability and the impact of control. In *ACM SIGCOMM*, August 1999.

[14] J. Ridoux, A. Nucci, and D. Veitch. Seeing the difference in IP traffic: Wireless versus wireline. In *IEEE Infocom*, April 2006.

[15] D. Veitch and P. Abry: Matlab code for the wavelet based analysis of scaling processes, http://www.cubinlab.ee.unimelb.edu.au/∼darryl/.

[16] P. Abry, M. S. Taqqu, P. Flandrin, and D Veitch. *Self-Similar Network Traffic and Performance Evaluation*, chapter Wavelets for the analysis, estimation, and synthesis of scaling data. Wiley, 2000.

[17] K. V. Vishwanath and A. Vahdat. Realistic and responsive network traffic generation. In *ACM SIGCOMM*, September 2006.