Modeling Multifractal Traffic with Stochastic L-Systems

Paulo Salvador, António Nogueira, Rui Valadas
Institute of Telecommunications / University of Aveiro, Portugal
e-mail: salvador@av.it.pt, nogueira@av.it.pt, rv@det.ua.pt

Abstract—**This paper proposes a novel multifractal traffic model, and an associated parameter fitting procedure, based on stochastic L-Systems, which were introduced by biologist A. Lindenmayer as a method to model plant growth. We provide a detailed comparison with a related multifractal model based on conservative cascades. Our results, that include applying the fitting procedure to real observed data with multifractal scaling behavior, show that L-System based models can achieve excellent fitting performance in terms of first and second order statistics and queuing behavior.** *

Keywords—**Traffic modeling, Self-similar, Multifractal, L-System.**

I. Introduction

Recent analysis of measured Internet WAN traffic has revealed that multifractal structures, such as random cascades, can help explaining the scaling behavior typically associated to networking mechanisms operating on small time scales (e.g. TCP flow control). A cascade (or multiplicative process) is a process that fragments a set into smaller and smaller components according to a fixed rule, and at the same time fragments the measure of the components by another (possibly random) rule. Random cascades were introduced by Mandelbrot as a physical model for turbulence [1]. In the traffic modeling context, the set can be interpreted as a time interval and the measure as the number of arrivals or number of bytes (in that interval).

The multifractal nature of network traffic was first noticed by Riedi and Lévy Véhel [2]. Subsequently various studies have addressed the characterization and modeling of multifractal traffic, essentially within the framework of random cascades [3] [4] [5] [6] [7] [8] [9]. In this paper, we propose a novel multifractal traffic model, and an associated fitting procedure, based on stochastic Lindenmayer-Systems (hereafter referred to as L-Systems). L-Systems are string rewriting techniques which were introduced by biologist A. Lindenmayer in 1968 as a method to model plant growth[10]. They are characterized by an alphabet, an axiom and a set of production rules. The alphabet is a set of symbols; the production rules define transformations of symbols into strings of symbols; starting from an initial string (the axiom), an L-System constructs iteratively sequences of symbols replacing each symbol by the corresponding string according to the production rules. Stochastic L-Systems are a method to construct recursively random sequences with multifractal behavior [11]. When compared with random cascades, stochastic L-Systems introduce a dependence on the construction process (due to the production rules), that has a meaningful physical explanation, and can help understanding the joint impact of network mechanisms and resource limitations on observed traffic. The proposed traffic model also includes the ability of modeling multiple scaling behavior.

This paper is organized as follows. In section II we give some background on L-Systems. In sections III and IV we present the traffic model and describe the associated fitting procedure. Section V provides a detailed comparison with random cascades and includes a physical explanation supporting L-System based traffic models. In section VI we discuss the results of applying the proposed fitting procedure to measured traffic traces. Finally, section VII presents the conclusions.

II. L-Systems Background

The basic idea behind L-Systems is to define complex objects by successively replacing parts of a simple object using a set rules. The L-System is a feedback machine that operates on strings of symbols. The set of symbols is called the alphabet. Starting from an initial state (called axiom), an L-System operates, at each iteration, by applying the set of production (or rewriting) rules simultaneously to all symbols of an input string to give an output string. For a comprehensive introduction to L-Systems see [11].

Consider a simple example of an organism growing through cell subdivisions. There are two types of cells represented by letters A and B. Cell subdivisions are modeled by replacing these symbols with strings of symbols: cell A subdivides into two cells represented by string AB; cell B subdivides into two cells represented by string AA. The ordering of the symbols is relevant in an L-System. The organism modeled by this L-System grows by repeated cell subdivisions. At birth the organism is the single cell A. After one subdivision the organism is two cells represented by string AB. After two subdivisions, the organism has four cells given by string ABAA, and after three subdivisions the organism has eight cells represented by string ABAAABAB. Using the formalism of L-Systems this growth process can be described as:

Alphabet:	$\{A,B\}$
Axiom:	A
Rules:	$A \rightarrow AB$
	$B \rightarrow AA$

The production rules can be stochastic. In stochastic L-Systems there may be several production rules for one symbol, and the specific rule is selected according to a probability distribution. Taking previous example, one production rule

*This work was part of project POSI/CPS/42069/2001 ”Traffic Modeling and Performance Evaluation of Multiservice Networks”, funded by Fundação para a Ciência e Tecnologia, Portugal. P. Salvador wishes to thank Fundação para a Ciência e a Tecnologia, Portugal, for support under grant BD/19781/99.

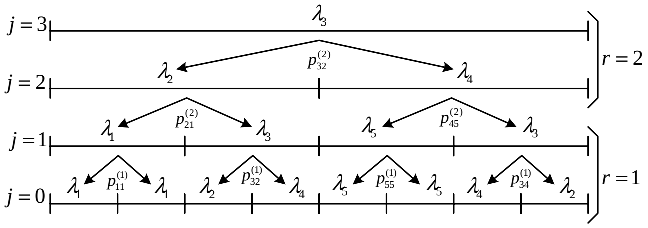

Fig. 1. Construction of an L-System based traffic model.

could be to convert A into AB with probability 0.4 or into BB with probability 0.6 (instead of converting always A into AB). In this case, after 3 iterations several strings are possible, e.g., ABAAABAB, ABABBBAB, or AAAAABBB. Stochastic L-Systems are a method to construct recursively random sequences with multifractal behavior [11].

III. Traffic Model

In this section we describe the proposed multifractal traffic model based on a stochastic L-System construction. We work on an alphabet of arrival rates defined by

$$\vec{\lambda} = \{\lambda_1, \lambda_2, ..., \lambda_L\}, \lambda_i \in I\!R_0^+, i = 1, ..., L. \quad (1)$$

and with production rules that randomly generates two arrival rates from a previous one. Without loss of generality, we assume $\lambda_1 < \lambda_2 < ... < \lambda_L$.

The traffic process is constructed progressively, governed by an L-System machine, where each iteration produces a new time scale. Starting with the coarsest time scale, where traffic is characterized by a single arrival rate over a single time interval, each iteration generates a finer time scale by (i) division of each (parent) time interval in two new equal length (child) subintervals and (ii) association of arrival rates to each new subinterval according to the production rules of the stochastic L-System. We allow the grouping of time scales in time scale ranges and the definition of different sets of production rules for each time scale range. This is motivated by the fact that each set of production rules maps into a distinct scaling behavior [12]. Classically, there will be scaling if the log-log plot of the q^{th} order energies (usual energy is $q = 2$) as a function of scale behaves linearly; if the plot is (globally) non-linear, different time scale ranges can be detected where linearity is observed (see [5], for example). The traffic process construction is illustrated in Figure 1.

To characterize the traffic process we define $X_{(j,r)}^{(i)} \epsilon \vec{\lambda}$ as the arrival rate at time interval i of time scale j and time scale range r. Let the number of scales be S and the number of ranges of scales be R. For convenience, we let j decrease from $j = S - 1$ (at the coarsest time scale) to $j = 0$ (at the finest time scale). Also, we let r decrease from $r = R$ (the range of coarsest time scales) to $r = 1$ (the range of finest time scales). Thus, the number of time intervals at time scale j, which we will denote by N_j, is 2^{S-j-1}. Moreover, assuming a unitary width for the intervals of the finest time scale, $j = 0$, the width at scale j will be 2^j. To relate time scales and time scale ranges we define j_r as the coarsest scale j in range r. Thus, in Figure 1, $S = 4$, $R = 2$, $j_2 = 3$ and $j_1 = 1$.

In order to assure that the average arrival rate is the same in all time scales, so as to maintain physical meaning, we will impose the following condition to the production rules:

$$X_{(j,r)}^{(i)} = \frac{1}{2} X_{(j-1,r')}^{(2i-1)} + \frac{1}{2} X_{(j-1,r')}^{(2i)} \quad (2)$$

i.e., the mapping of arrival rates is such that the arrival rate averaged over the left and right child subintervals will be equal to the parent arrival rate. With this condition, the traffic process generation can be described by axiom $X_{(S-1,R)}^{(1)}$, the arrival rate at the coarsest time scale, and production rules defined by

$$X_{(j,r)}^{(i)} = \lambda_l \xrightarrow{p_{lq}^{(r)}} \begin{cases} X_{(j-1,r')}^{(2i-1)} = \lambda_q \\ X_{(j-1,r')}^{(2i)} = 2\lambda_l - \lambda_q \end{cases} \quad (3)$$

where $\sum_{l=1}^{L} p_{lq}^{(r)} = 1, \forall q$. Thus, an arrival rate λ_l in interval i, scale j and range r produces, with probability $p_{lq}^{(r)}$, arrival rate λ_q at the left subinterval $2i-1$ and arrival rate $2\lambda_l - \lambda_q$ at the right subinterval $2i$, of next scale $j-1$ and range r'. The production rules can be totally described by R $L \times L$ matrices

$$\mathbf{P}^{(r)} = \left(p_{lq}^{(r)}\right), \quad l, q = 1, ..., L, \quad r = 1, ..., R \quad (4)$$

In order to guarantee that the alphabet is closed with respect to the production rules we impose the following conditions: (i) $\lambda_i - \lambda_{i-1} = \frac{\lambda_L - \lambda_1}{L-1}, i = 2, 3, ..., L$, i.e., the λ_i values will be equidistant; (ii) $p_{lq}^{(r)} = 0$ if $q > l + \min(L-l, l-1)$ or $q < l - \min(L-l, l-1)$.

Finally, the L-System construction defines, at scale j and range r, the sequence

$$Y_{(j,r)} = \{X_{(j,r)}^{(i)}, i = 1, ..., N_j\} \quad (5)$$

IV. Fitting Procedure

The fitting procedure determines the L-System parameters from real data observations. It starts by fixing a sampling interval Δ and considering the time series representing the total number of packet arrivals in each non-overlapping sampling interval. Let this (empirical) time series be $\{A_k, k = 1, 2, ..., K\}$, where A_k represents the number of arrivals in sampling interval k. For convenience, we take the length of the time series K to be a power of 2. The inference procedure can then be divided in three steps: (i) determination of the L-System alphabet and axiom, (ii) identification of the time scale ranges and (iii) inference of the L-System production rules.

The alphabet of the L-System will consist in L equidistant arrival rate values, ranging from the minimum to the maximum values present in data. The value of L is a compromise between accuracy and complexity. Due to the mass preservation property of L-Systems defined in (2), the axiom is inferred as the average arrival rate of $\{A_k\}$, rounded to the closest alphabet element, i.e.,

$$X^{(1)}_{(S-1,R)} = \Lambda\left((1/K\Delta)\sum\nolimits_{k=1}^{K} A_k\right) \tag{6}$$

where $\Lambda(x)$ represents a function that rounds x towards the nearest element of $\vec{\lambda}$.

The identification of time scale ranges is based on wavelet scaling analysis. We use the method described in [5], which resorts to the (second-order) logscale diagram. A (second-order) logscale diagram is a plot of y_j against j, together with confidence intervals about the y_j, where y_j is a function of the wavelet discrete transform coefficients at scale j. The time scale ranges correspond to the set of time scales for which, within the limits of the confidence intervals, the y_j fall on a straight line, i.e., the scaling behavior is linear in a time scale range. Figure 3 shows the logscale diagram of a trace measured at the University of Aveiro (which is described in section VI). There are 4 time scale ranges (within a total of 18 time scales) defined by $j_1 = 3, j_2 = 7, j_3 = 10$ and $j_4 = 17$.

The final step is the inference of the L-System production rules, which are fully characterized by the $\mathbf{P}^{(s)}$ matrices. First, data is rounded in order to define sequence $Y_{(j,r)}$ at each time scale. This comprises obtaining the arrival rates $X^{(i)}_{(j,r)}$ from $\{A_k\}$ through

$$X^{(i)}_{(j,r)} = \Lambda\left((N_j/K\Delta)\sum\nolimits_{k=K(i-1)/N_j+1}^{Ki/N_j} A_k\right) \tag{7}$$

with $i = 1, ..., N_j$, for each j. Letting $c^{(r)}_{lq}$ represent the number of times that, at scale j and range r, the parent $X^{(i)}_{(j,r)} = \lambda_l$ produced the left child $X^{(2i-1)}_{(j-1,r')} = \lambda_q$, the production rule probabilities can be inferred as

$$p^{(r)}_{lq} = c^{(r)}_{lq} / \sum_{q=1}^{L} c^{(r)}_{lq}, \quad l = 1, ..., L, \quad r = 1, ..., R \tag{8}$$

V. Relation with Random Cascades

A random cascade is characterized by an initial mass, uniformly distributed over a single interval, which is subdivided along the different stages of the cascade construction. Each interval is divided in two (or more) identical subintervals and the mass is randomly assigned to each subinterval, according to a random variable W called the generator. In the case of a traffic process the mass can be interpreted as the number of arrivals or bytes within a time interval. Let $W_{i,j}, i = 1, ..., N_j, j = S-1, ..., 0$, denote an (independent) random variable, having the same distribution as W, that redistributes mass from time scale j into time interval i (belonging to subsequent time scale $j-1$). The construction of the traffic process, starts at the coarsest time scale $S-1$ with an initial mass M distributed uniformly over a unit time interval. We will restrict our discussion to the case where a (parent) interval is subdivided in only two (child) intervals. In the first iteration, a finer time scale is produced, by dividing the time interval in two new subintervals of length $1/2$, and assigning mass $MW_{1,S-1}$ to the left subinterval and $MW_{2,S-1}$ to the right subinterval. At the second iteration, producing time scale $S-2$, each of these intervals generates new subintervals, one at the left and another at the right, giving rise to four subintervals of length $1/4$ with masses $MW_{1,S-1}W_{1,S-2}$, $MW_{1,S-1}W_{2,S-2}$, $MW_{2,S-1}W_{3,S-2}$ and $MW_{2,S-1}W_{4,S-2}$, respectively. Iterating this construction process, the mass of parent interval i from scale j is redistributed into child subintervals $2i-1$ and $2i$ of scale $j-1$ with probabilities $W_{2i-1,j}$ and $W_{2i,j}$, respectively. Note that the mass at each time scale is only preserved in expectation. Note also that the way mass gets redistributed is independent from mass itself, since the $W_{i,j}$ are independent of mass.

Feldmann et al. [3] have proposed the use of conservative cascades, a special case of random cascades, as a model for IP traffic. In conservative cascades, the generator W takes on values in $(0,1)$, has mean $1/2$ and is symmetric about its mean. Furthermore, mass is redistributed such that the total mass assigned to left and right child subintervals remains equal to that of the parent interval. Thus, the mass is preserved throughout the splitting process: if the mass of the i^{th} parent interval is Q, the mass of the left child interval will be $QW_{2i-1,j}$ and that of the corresponding right child interval will be $Q(1 - W_{2i-1,j})$. As a result, the mass at all stages will be (exactly) M, if the initial mass is M. This is the mass preservation property which also occurs in our stochastic L-System construction. Feldmann et al. supported the adoption of conservative cascade models in the networking context by observing that the transmitted traffic is constructed through fragmentation at successive network layers, and that the total number of bytes is roughly preserved during this fragmentation process. A typical example is the dynamics of a Web session, where user clicks results in requests, requests give rise to connections, connections are made up of flows, and flows consist of individual packets.

In L-Systems, the mass in interval i of scale j is $2^j X^{(i)}_{(j,r)}$. In addition to the mass preservation property, L-Systems include another important feature that is not present in conservative cascades (nor in random cascades) and has a meaningful physical explanation. In L-Systems the way mass is redistributed to left and right child subintervals can be made dependent on the mass of the parent interval, whereas in the conservative cascade construction this dependence is not allowed. Recall that a parent arrival rate λ_l gives rise to a left child arrival rate λ_q with probability $p^{(r)}_{lq}$, which depends explicitly on l. Take again the example of Feldmann et al. and consider the way Web requests are scheduled over time. The number of requests per time interval that a Web server can handle is limited, due to resource availability constraints. Therefore, the way user clicks produce requests depends on the overall number of clicks. If a Web server has more requests to process it will distribute them more sparsely over time. Thus, the mass itself (the number of clicks, in this case) influences the way distribution takes place. At a lower level, consider the way flows produce packets. If the network is more congested, the feedback control exercised by TCP imposes that packets will be more sparsely distributed over time. Thus, once more, the mass (the number of flows, in this case) influences the way distribution takes place.

To bring further insight into this issue, we next evaluate whether or not the independence property of the generator, subjacent to conservative cascades, is present in data, using the pOct.TL Bellcore trace. The generator, as defined in the context of conservative cascades, represents the fraction of mass distributed to a left child subinterval. In L-Systems this fraction is $2^{j-1}X^{(2i-1)}_{(j-1,r')}/2^jX^{(i)}_{(j,r)} = \lambda_l/2\lambda_q$. We define $W_{j|l}$ as the generator at time scale j conditioned on the arrival rate λ_l. Thus, the $W_{j|l}$ are discrete random variables that, for each l, can take values $\lambda_l/2\lambda_q$, and can be easily inferred from data. Clearly, if the independence assumption is true then, at each time scale j, the $W_{j|l}$ should have the same distribution. In Figure 2 we plot the variance of the inferred conditional $W_{j|l}$, for all λ_l and for j = 1, 2, 3, 4 and 5. Results show that, within the same time scale, the generators can have very different variances (the same is true for the mean). For example, at $j = 0$ the variance is 0.04 for $\lambda_l = 21$ pkts/sec and is 0.001 for $\lambda_l = 100$ pkts/sec. Therefore, they have dissimilar distributions which indicates that the independence assumption is not valid in the case of the pOct.TL Bellcore trace. We have observed this same behavior in many other traces. To conclude, the effect of resource availability limitations is an important factor with strong impact in the traffic generation process, which can be captured through an L-System based construction (but not through a conservative cascade).

VI. Numerical Results

We have applied our fitting procedure to two traces of IP traffic: (i) the well known pOct.TL Bellcore trace [13] and (ii) one trace measured at the University of Aveiro (UA) which exhibits non-trivial multifractal scaling behavior. The sampling interval was 0.1 seconds in both traces. The UA trace is representative of Internet access traffic produced within a University campus environment. The University of Aveiro is connected to the Internet through a 10 Mb/s ATM link and the measurements were carried out in a 100 Mb/s Ethernet link connecting the border router to the firewall, which only transports Internet access traffic. The UA trace consists of 20 millions packets captured on July 3^{rd} 2001 from 8.00 pm to 3.15 am. The traffic analyzer was a 1.2 GHz AMD Athlon PC, with 1.5 Gbytes of RAM, running WinDump. The measurements recorded the arrival instant and the IP header of each packet. The mean arrival rate of the UA trace is 766 pkts/sec.

The pOct.TL trace was fitted to a stochastic L-System with an alphabet of $L = 243$ arrival rates, from the minimum to the maximum present in data, in steps of 10 pkts/sec (the minimum and maximum were 10 pkts/sec and 2430 pkts/sec, respectively). The logscale diagram identified 5 time scale ranges (within a total of 14 time scales) defined by $j_1 = 3, j_2 = 6, j_3 = 8, j_4 = 9$ and $j_5 = 13$. In the case of the UA trace the alphabet size was $L = 469$, from 170 pkts/sec to 4850 pkts/sec in steps of 10 pkts/sec. As referred before, the UA trace has 4 time scale ranges (Figure 3). The parameter estimation took less than 30 seconds, using a MATLAB implementation running in the PC described above. This shows that the fitting procedure is computationally very efficient (note that the size of the alphabet, the number of ranges and the size of the trace, which determine the computational time, are all relatively large).

We assess the suitability of the traffic model and the accuracy of the fitting procedure using several criteria. We compare both the probability and autocovariance functions of the packet counts (number of packet arrivals in sampling interval) obtained with the fitted stochastic L-System model and with the original data trace. We also analyze the queuing behavior by comparing the packet loss ratio, obtained through trace-driven simulation, using two types of input traffic: (i) the original trace and (ii) the trace generated according to the fitted stochastic L-System. Our comparisons are extended to a conservative cascade model. The model was inferred using the procedure presented in [7]. Here, the generator at each time scale is fitted to a truncated normal distribution, where the mean is always $1/2$ but the variance is adjusted individually at each time scale.

Multifractality is assessed using a linear multiscale diagram [5]. In this case, multifractal scaling behavior is detected when there is no horizontal alignment (within the limits of confidence intervals). Figure 4 shows that the UA trace, and the traces generated using both the L-System and conservative cascade models, all have multifractal scaling behavior. A similar analysis carried out on the pOct.TL trace revealed that this trace is not multifractal.

In the case of the probability function (Figure 5), both the conservative cascade and the L-System models fitted very well the UA trace. The performance of the conservative cascade model was not so good in the case of the pOct.TL trace. This can be explained by the symmetry imposed on the generator W, which makes it difficult for conservative cascades to fit asymmetric probability distributions. In the case of the autocovariance function (Figure 6), the fitting performance achieved by both models was again very good in the case of the UA trace, but clearly inferior for conservative cascades in the case of the pOct.TL trace.

To assess the queuing behavior the buffer size was varied from 10 Kbytes to 2 MBytes. The service rate was 501.2 KBytes/s for the pOct.TL trace (corresponding to an utilization of 0.7) and 572 KBytes/s for the UA trace (corresponding to an utilization of 0.8). Figure 7 shows that, for both traces, the fitting of the queuing behavior was very good in the case

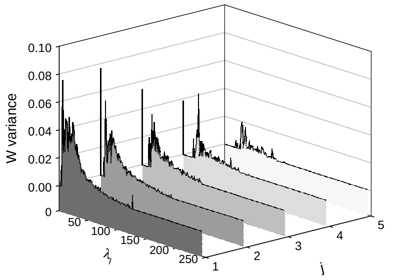

Fig. 2. $W_{j|l}$ variance versus λ_l and j, trace pOct.TL.

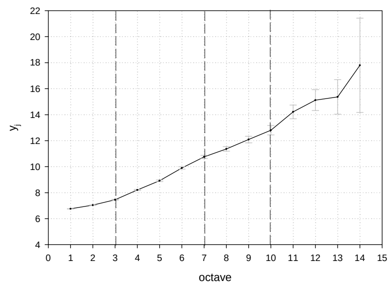

Fig. 3. Logscale diagram, trace UA.

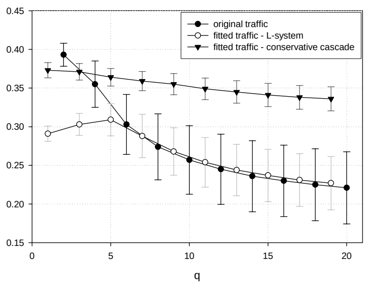

Fig. 4. Linear multiscale diagram, trace UA.

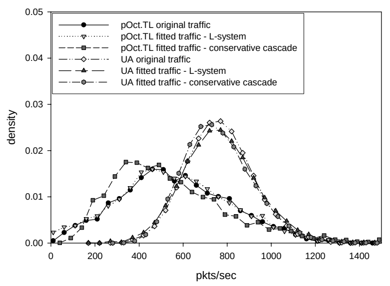

Fig. 5. Probability function of packet counts, traces pOct.TL and UA.

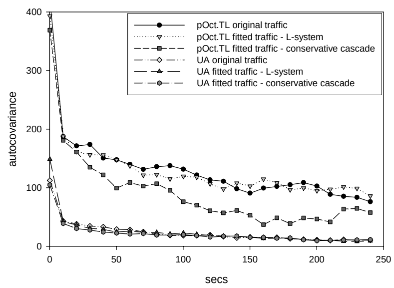

Fig. 6. Autocovariance of packet counts, traces pOct.TL and UA.

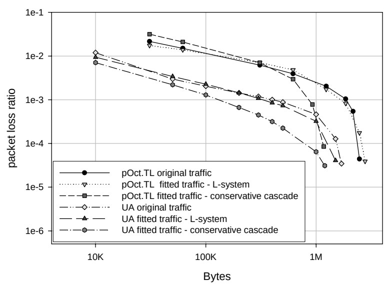

Fig. 7. Packet loss ratio versus buffer size, traces pOct.TL and UA.

of L-Systems but significant differences occurred with conservative cascades. It is interesting to note that, in the case of the UA trace, these differences occurred despite of a very good fitting in the first and second order statistics. This clearly illustrates that first and second order statistics are not sufficient to characterize multifractal traffic.

We also have analyzed a number of other traces. Our results show that, in general, L-Systems achieves better performance than conservative cascades. This can be explained as follows. First, L-Systems allow the mass redistribution to depend on the mass itself, a feature that is clearly present in real observed data. Second, L-Systems provide a higher number of parameters, which are meaningful from the point of view of physical reality. Third, the generator in conservative cascades is assumed to be symmetric which restricts the fitting of the probability function.

VII. Conclusions

This paper proposed a novel multifractal traffic model, and an associated parameter fitting procedure, based on stochastic L-Systems, which were introduced by biologist A. Lindenmayer as a method to model plant growth. We provided a detailed comparison with a related multifractal model based on conservative cascades. Our results, that include applying the fitting procedure to real observed data with multifractal scaling behavior, showed that L-System based models can achieve excellent fitting performance in terms of first and second statistics and queuing behavior.

References

[1] B. Mandelbrot, "Intermittant turbulence in self.similar cascades: Divergence of high moments and dimensions of the carrier," *Journal of Fluid Mechanics*, vol. 62, pp. 331–358, 1974.

[2] R. Riedi and J. L. Véhel, "Multifractal properties of TCP traffic: a numerical study," *Technical Report No 3129, INRIA Rocquencourt, France*, Feb. 1997, Available at www.dsp.rice.edu/∼riedi.

[3] A. Feldmann, A.C. Gilbert, and W. Willinger, "Data networks as cascades: Investigating the multifractal nature of internet WAN traffic," in *Proceedings of SIGCOMM*, 1998, pp. 42–55.

[4] R. Riedi, M. Crouse, V. Ribeiro, and R. Baraniuk, "A multifractal wavelet model with application to network traffic," *IEEE Transactions on Information Theory*, vol. 45, no. 4, pp. 992–1018, April 1999.

[5] P. Abry, P. Flandrin, M. Taqqu, and D. Veitch, "Wavelets for the analysis, estimation and synthesis of scaling data," *in Self-Similar Network Traffic Analysis and Performance Evaluation, K. Park and W. Willinger, Eds.*, 1999.

[6] J. Gao and I. Rubin, "Multifractal analysis and modeling of long-range-dependent traffic," in *Proceedings ICC'99*, June 1999, pp. 382–386.

[7] A. Gilbert, W. Willinger, and A. Feldmann, "Scaling analysis of conservative cascades, with applications to network traffic," *IEEE Transactions on Information Theory*, vol. 45, no. 3, pp. 971–992, April 1999.

[8] A. Erramilli, O. Narayan, A. Neidhardt, and I. Saniee, "Performance impacts of multi-scaling in wide area TCP/IP traffic," in *Proceedings of INFOCOM'2000*, 2000.

[9] D. Veitch, P. Abry, P. Flandrin, and P. Chainais, "Infinitely divisible cascade analysis of network traffic data," in *Proceedings of the Int. Conf. on Acoust. Speech and Sig. Proc.*, June 2000.

[10] A. Lindenmayer, "Mathematical models for cellular interactions in development II. Simple and branching filaments with two-sided inputs," *Journal of Theoretical Biology*, vol. 18, pp. 300–315, 1968.

[11] H. Peitgen, H. Jurgens, and D. Saupe, *Chaos and Fractals: New Frontiers of Science*, Springer-Verlag, 1992.

[12] P. Salvador, A. Nogueira, and R.Valadas, "Scaling behavior analysis of multifractal traffic models based on stochastic L-systems," Tech. Rep., University of Aveiro, 2002.

[13] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, Feb. 1994.