ICT networks, network traffic,
fractal structure

WŁODARSKI Przemysław[1]

ICT NETWORK TRAFFIC BASED ON A SIMPLE FRACTAL STRUCTURE

The results of measurements of ICT network traffic reveals the presence of fractal structures and long-range dependence. Accurate modeling of the offered traffic load is the first step in optimizing resource allocation algorithms and QoS requirements. In this article the fractal structure is represented by the simple random midpoint displacement method which approximates fractional Brownian motion (fBm), theoretical process that exhibits aforementioned features, is presented. Apparently, this algorithm generates inaccurate approximation of fBm for the higher degree of self-similarity. To improve this simple and fast algorithm, some modifications based on variance analysis are proposed. The results of simulation and statistical testing with comparison to the real-time measurements in computer network are presented and discussed.

NATĘŻENIE RUCHU W SIECIACH TELEINFORMATYCZNYCH OPARTE NA PROSTEJ STRUKTURZE FRAKTALNEJ

Wyniki pomiarów natężenia ruchu w sieciach teleinformatycznych ujawniają obecność struktur fraktalnych oraz zależności długoterminowych. Dokładne modelowanie obciążenia stanowi zatem pierwszy krok w optymalizowaniu rezerwacji zasobów sieciowych oraz pomaga spełnić wymagania QoS (Quality of Service). W artykule strukturę fraktalną otrzymano za pomocą metody losowego przemieszczenia środka odcinka, dzięki czemu otrzymano przybliżenie procesu ułamkowego ruchu Browna stanowiącego teoretyczną podstawę w analizie struktur fraktalnych. Jak się okazuje, zaprezentowana metoda niezbyt dokładnie przybliża ten teoretyczny proces, szczególnie dla wyższych stopni samopodobieństwa. Aby ulepszyć tą prostą i szybką metodę zaproponowano pewne modyfikacje bazujące na analizie wariancji. Zaprezentowano wyniki symulacji oraz testowanie statystyczne w zestawieniu porównując je do rzeczywistego ruchu zarejestrowanego w sieci teleinformatycznej.

[1] West Pomeranian University of Technology, Faculty of Electrical Engineering, Department of Signal Processing and Multimedia Engineering, 71-126 Szczecin, ul. 26 Kwietnia 10, tel.: +4891 449-5311, fax: +4891 449-5347, e-email: Przemyslaw.Wlodarski@zut.edu.pl

1. INTRODUCTION

1.1. Fractal structure in the ICT network traffic and fractional Brownian motion

Packet arrivals in ICT networks was often assumed to be Poisson process because it has attractive theoretical properties [1,2]. Studies shows that local-area and wide-area network traffic is much better modeled using self-similar processes, which have much different theoretical properties than Poisson process [2,3,5] (Fig.1). A statistical analysis of Ethernet traffic reveals the presence of “burstiness” across a wide range of time scales which is best explained in terms of self-similarity, i.e., traffic patterns show structural similarities across a wide range of time scales.

Below, on Fig.1 there is a sample traffic patterns in a different time scales. Picture show packet arrivals of incoming traffic captured on the main firewall of the West Pomeranian University of Technology as well as Poisson model. We can see that patterns for real traffic are far more “bursty” than for Poisson model, especially as the time scale increases. The degree of fractal level (self-similarity) both for real-time traffic measurements and presented model will be considered further.

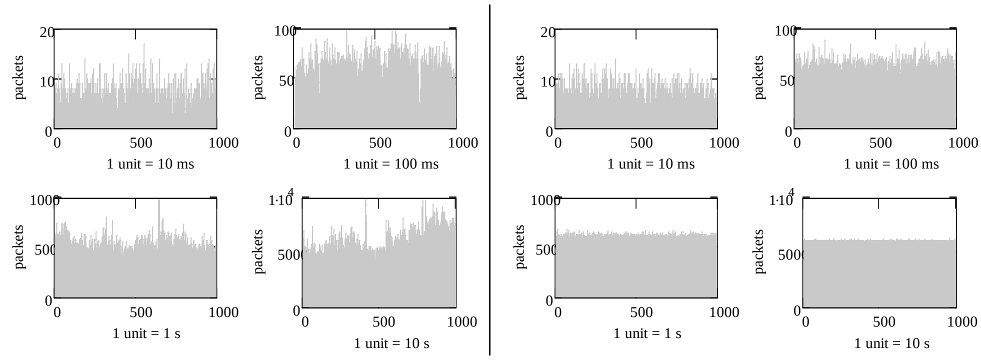


Fig. 1. Packet traffic (left) and Poisson model (right) patterns in real ICT network across different time scales

As it was mentioned earlier, fractional Brownian motion (fBm) is the theoretic exactly self-similar process with stationary increments and has the following properties [4,6]:

- invariance in distribution – statistical self-similarity

$$Y(at) = a^H Y(t) \quad , \quad t, a \in R \,, \; a > 0 \tag{1}$$

- has the stationary Gaussian increments with the density distribution ($\mu = 0$):

$$f_G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{2}$$

- has the covariance function defined as:

$$Cov(Y(s),Y(t)) = K_H(s,t) = \frac{\sigma^2}{2}\left(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\right) \quad , \quad t,s \in R \tag{3}$$

- for $\sigma > 0$ and $0.5 \le H < 1$

$$E\left(Y^2(t)\right) = \sigma^2 |t|^{2H} \tag{4}$$

The index H is called the Hurst exponent and is the measure of the degree of self-similarity ($0.5 \le H < 1$). When $H = 0.5$, the fBm is the usual Brownian motion with the following relationship:

$$Cov(Y(s),Y(t)) = K_{0.5}(s,t) = \min(s,t) \tag{5}$$

1.2. Simple fractal structure

Simple fractal structure represented by the random midpoint displacement algorithm is best described in [7]. This very simple and powerful algorithm is used for example in fractal interpolating or in modeling computer landscapes.

We look for the real values of $X(t)$ for $0 \le t \le 1$. First, let $X(0) = 0$ and $X(1)$ is the Gaussian random number. Then we calculate $X(1/2)$ as the mean of $X(0)$ and $X(1)$. In the next step we add the correction D_1 which is the Gaussian random number. D_1 should be multiplied by the scale parameter as in (6). In the next step we reduced the scale parameter by 2^H and divide the time segment $[0;1]$ into $[0;1/2]$ and $[1/2;1]$. So, $X(1/4)$ equals $1/2\left(X(0) + X(1/2)\right)$ plus the random correction D_2. For $X(3/4)$ we have: $X(3/4) = 1/2(X(1/2) + X(1)) + D_2$. The procedure repeats until we get satisfactory level of division.

The algorithm produces 2^J real values of $Y(t)$, where J is the level of division of time segment $[0;1]$. Scale parameters for corrections $D_1, D_2, ..., D_J$ can be obtained by calculating the variance, i.e.:

$$Var\left[X(1/2)\right] = Var\left[1/2\left(X(0) + X(1)\right) + D_1\right]$$

$$Var(D_1) = \sigma^2 \frac{1 - 2^{2H-2}}{2^{2H}} \tag{6}$$

So, the correction for D_1 is $\sqrt{1 - 2^{2H-2}} \cdot 2^{-H}$. For $\{D_j\}$, $j = 1,2,...,J$ we have

$$Var\left(D_j\right)=\sigma^2\,\frac{1-2^{2H-2}}{2^{j2H}} \tag{7}$$

2. SIMULATION AND PROPOSED IMPROVEMENT

2.1. Simulation results

Assuming $\sigma=1$ we expect to get the approximation of fBm which has the standard Gaussian distribution $N(0,1)$ for different H . Because of time of computation we take 3000 paths of $N=30$ different H values. Thus we need $M=100$ different standard Gaussian random sequences, each consisted of 2^J samples. We take into consideration J = 9, 10, 11 that corresponds to I = 512, 1024, 2048 samples per each random sequence. If we check variances of the increment process for each H of M generated path, we can see that for the higher values of H (above approx. 0.7) variances monotony decreases almost to 0 (*Fig.2*). This phenomenon causes that we do not get standard random Gaussian numbers for higher values of *H*.

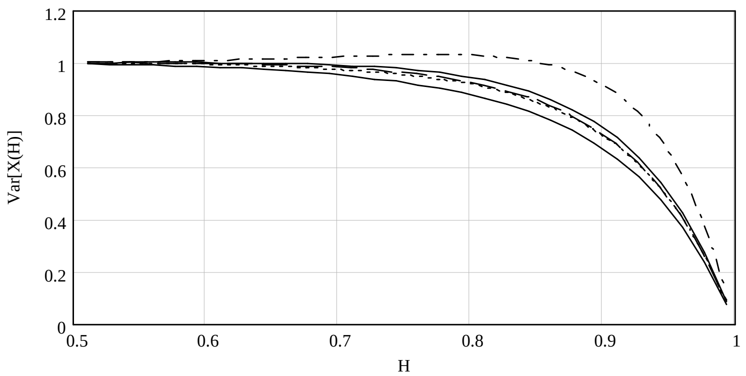


Fig. 2. Variances of 5 sample sequences for different H values

2.2. Improvement

Since variances of generated sequences using above mentioned algorithm behave quite regular, we can try to make additional correction for generated values approximating $F_V(H,m)=Var\left[X^{(m)}(H)\right]$ by the function that best fits to $F_V(H,m)$ for all $m=1,2,...,M$ and depends on H and on one extra parameter, say c_m ($F_p(H,c_m)$). The correction:

$$X_p^{(m)}(H)=X^{(m)}(H)\cdot F_p(H,c_m)^{-0.5} \tag{8}$$

should eliminate this unwanted behavior of variance. We test 3 different functions:

$$f_1(H,c) = \frac{1-H^c}{1-0.5^c} \tag{9}$$

$$f_2(H,c) = \frac{2}{\pi}\cdot \text{arctg}\left[c\cdot\left(\frac{1}{H-0.5}-2\right)\right] \tag{10}$$

$$f_3(H,c) = \text{tgh}\left[c\cdot\left(\frac{1}{H-0.5}-2\right)\right] \tag{11}$$

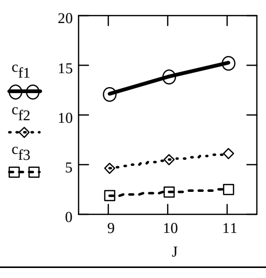
In order to find c parameter we use mean-square error (MSE) technique, so we have to solve the following equations:

$$\sum_{n=0}^{N-1}\left[Var_n - \frac{1-(H_n)^c}{1-0.5^c}\right]\cdot\left[\frac{(H_n)^c\cdot\ln(H_n)\cdot(1-0.5^c)-0.5^c\cdot\ln(0.5)\cdot\left[1-(H_n)^c\right]}{(1-0.5^c)^2}\right]=0 \tag{12}$$

$$\sum_{n=0}^{N-1}\left[Var_n - \frac{2}{\pi}\cdot\text{arctg}\left[c\cdot\left(\frac{1}{H_n-0.5}-2\right)\right]\right]\cdot\frac{-\frac{2}{\pi}\cdot\left(\frac{1}{H_n-0.5}-2\right)}{1+\left[c\cdot\left(\frac{1}{H_n-0.5}-2\right)\right]^2}=0 \tag{13}$$

$$\sum_{n=0}^{N-1}\left[Var_n-\text{tgh}\left[c\cdot\left(\frac{2-2\cdot H_n}{H_n-0.5}\right)\right]\right]\cdot\left[1-\text{tgh}\left[c\cdot\left(\frac{2-2\cdot H_n}{H_n-0.5}\right)\right]^2\right]\cdot\left(\frac{2\cdot H_n-2}{H_n-0.5}\right)=0 \tag{14}$$

Tab. 1. Results for estimating c parameter in functions (9), (10) and (11)

	Mean *c* value	Mean *MSE*			Coefficient of variation		
	J = 9,10,11	*J* = 9	*J* = 10	*J* = 11	*J* = 9	*J* = 10	*J* = 11
f_1		0.007	0.013	0.009	0.265	0.348	0.294
f_2		0.033	0.038	0.033	0.331	0.456	0.373
f_3		0.066	0.061	0.048	0.280	0.358	0.302

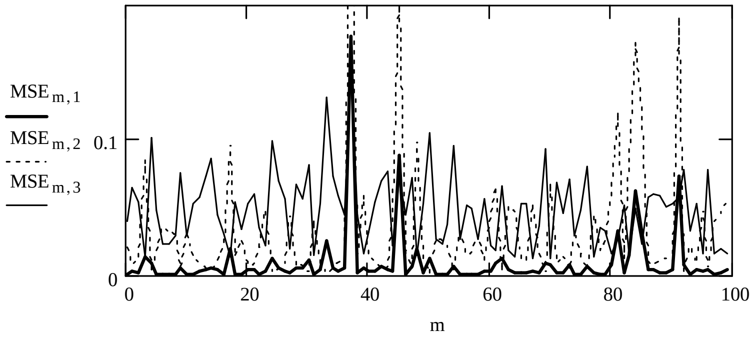


Fig. 3. Mean-square error for m-th random sequence

Looking at fig. 3, $MSE_{m,k}$, $k=1,2,3$ denotes the mean-square error of fitting $f_k(H,c_m)$ into $F_V(H,m)$ for all $m=1,2,...,M$ .In the highlighted region of Table 1 there are the smallest values of mean MSE and coefficients of variation from the proposed functions. In the next paragraph there are the results of statistical testing for the (9) function which best fits to the changing variance of $X^{(m)}(H)$.

3. STATISTICAL TESTING

3.1. Testing of the normality and variance

For normality testing the Kolmogorov's test was used. It compares both theoretical (standard Gaussian) and empirical cumulative distribution functions and verifies the hypothesis $H_0 : F_e = F_t$ against $H_1 : F_e \neq F_t$ [9]. The significance level is set to α=0.05 and f*ig.4* shows values of test statistic above 0.8. Values between 0.8 and 1 are marked with "+". Sequences that exceed 1 fail test and are marked with "o".

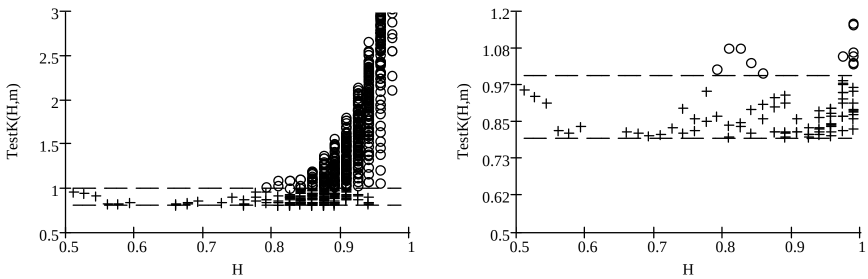


Fig. 4. Kolmogorov's tests for standard (left) and improved (right) RMD method (J=10)

For variance testing the F-Snedecor test was used. Results for each *J* tested shows increasing trend of failed test (tab. 2). This is normal phenomenon, because number of samples multiplies 2 in each step (512, 1024, 2048) and we can observe such tendency also for samples taken from any random variable generator.

Tab. 2. Random sequences that failed particular tests (all seq.: 3000)

	Algorithm	Kolmogorov's test	F-Snedecor test
J = 9 (512)	standard	614 (20.5 %)	941 (31.4 %)
	improved	0 (0 %)	6 (0.2 %)
J = 10 (1024)	standard	674 (22.5 %)	972 (32.4 %)
	improved	12 (0.4 %)	28 (0.9 %)
J = 11 (2048)	standard	726 (24.5 %)	1015 (33.8 %)
	improved	27 (0.9%)	62 (2.1 %)

3.2. Testing of the self-similar level

The standard and improved method have been tested for self-similarity. For testing the following methods have been used [2], [10]: aggregated variance, index of dispersion for counts and periodogram-based method. Fig. 5 shows an example of the aggregated variance method which compares data for real network traffic measured at the West Pomeranian University of Technology and data for improved method generator for H=0.806 (estimated value for the real traffic). Dashed line corresponds to the non-self-similar process (i.e. Poisson).

As it turns out, the Hurst parameter estimation for random sequences generated using standard random midpoint displacement method is exactly the same like for random sequences of improved algorithm - curves for each estimation method overlap.

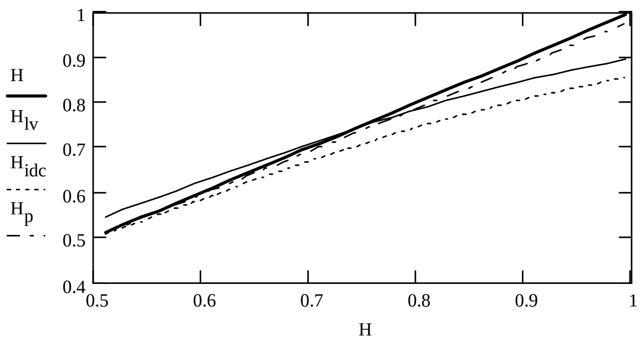


Fig. 5. Hurst parameter estimation (H – desired Hurst parameter) for 3 methods: Hlv - aggregated variance, Hidc - IDC, Hp- periodogram-based method

4. CONCLUSIONS

Method presented in this article that was used to generate approximate fBm is fast and easy to implement. The only drawback is that we do not get Gaussian distribution with desired σ parameter for higher values of *H* (especially greater than approx. 0.7). We could fix this inconvenience by the additional correction proposed in section 2.2. Three approximation functions have been tested. First of them (9) had the minimal mean-square error (tab. 1) and was chosen for the further analysis. Statistical testing shows a big advantage of the improved method (tab. 2). Very few (less than 1%) of sequences of thge proposed method failed Kolmogorov’s goodness-of-fit tests for standard normality (comparing to the less than 25% value for standard method). The advantage is also confirmed by variance testing using F-Snedecor statistic (only 2.1 % versus 33.8 % for the worst case when *J*=11). Furthermore, both standard and improved methods reveal self-similar properties for various *H* but the latter approximates fBm more accurately and thus can be used to model self-similar network traffic that has Gaussian distribution.

5. REFERENCES

[1] H. Elbiaze et al., ”A new structure-preserving method of sampling for predicting self-similar traffic”. Telecommunication Systems 43 265-277, 2010.
[2] W. Leland, M. Taqqu, W. Willinger, D. Wilson, “On the Self-Similar Nature of Ethernet Traffic”, *IEEE/ACM Transactions on Networking*, 2(1), pp. 1-15, 1994.
[3] W. Willinger, M. Taqqu, R. Sherman, D. Wilson, “Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level *IEEE/ACM Transactions on Networking*, 5(1), pp. 71-86, 1997.
[4] M. E. Crovella, A. Bestavros, “Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes”, *IEEE/ACM Transactions on Networking*, 5(6), pp. 835-846, December 1997.
[5] V. Paxson, S. Floyd, “Wide-area traffic: The Failure of Poisson Modeling”, *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226-244, 1995.
[6] P. Doukhan, G. Oppenheim, M. Taqqu, “Theory and Applications of Long-Range Dependence”, Birkhauser, 2002.
[7] H. O. Peitgen, H. Jurgens, D. Saupe, “Chaos and Fractals: New Frontiers of Science”, Springer Verlag, 1992.
[8] M. Taqqu, W. Willinger, R. Scherman. ”Proof of a Fundamental Result in Self-Similar Traffic Modeling”, *Computer Communication Review*, 27 (1997), pp. 5-23.
[9] L. Gajek, M. Kałuszka, “Wnioskowanie statystyczne”, WNT, 2000.