# A Tool for Long-memory Analysis in Computer Network Time Series

Julio C. Ramírez Pacheco[1], Deni Torres Román[2], and Joel Trejo Sanchez[1]

[1] Universidad del Caribe, SM 78, Manzana 1, lote 1, 77528, Cancún, QRoo, México
jramirez@ucaribe.edu.mx, jtrejo@ucaribe.edu.mx
[2] CINVESTAV Unidad Guadalajara, Av Científica 1145, 45010
Col. el Bajío, Zapopán, Jalisco, México
dtorres@gdl.cinvestav.mx

**Abstract.** The paper presents a novel software tool for *Hurst-index* estimation in *self-similar* and *long-range dependent* computer network time series. The tool, named *Variance Analyzer*, is based on the aggregated variance algorithm with tuned cut-offs. A comparison with *Selfis*, a similar tool for *long-memory*, using different dependence characteristics(*fGn* and *fDn*), shows that *Variance analyzer* presents better accuracy, time of convergence and faster estimations. Similar results are also obtained when using well-known real LAN traffic. The sources of inaccuracies in the algorithm are identified and the correct tuning is proposed.

## 1 Introduction

Computer network traffic's non-standard behavior is well studied and has been observed in several network configurations(LANs, WANs, VBR traffic, etc) [1] [2] [3] [4] [5]. *Extremes*, *heavy-tails*, *self-similarity* and *long-range dependence* are present in *delay*, *delay jitter*, file size, transmission times and aggregate traffic traces. The presence of these phenomena have a deleterious impact on computer networks' performance affecting the *quality of service* of applications [6] [9] [13]. Thus, an important problem in these traces is to correctly fit a model and then to efficiently quantify the degree of non-standard behavior(*tail-index* or *Hurst index* estimation) for the particular selected model. Once the model and the estimation is performed, the next step is to take actions in order to improve *quality of service* degree and the overall network performance. To accomplish the estimation, several algorithms have been proposed, each algorithm presents varying degrees of accuracy and time-domain or frecuency-domain properties [12] [7] [14] [13]. In this paper, a novel software tool for *self-similarity* and *long-range dependence* analysis in computer network time series is presented. The C++ based tool, named *Variance Analyzer*, is based on the time-domain aggregated variance algorithm. A comparison procedure against *Selfis*, a similar tool for *long-memory*, is accomplished. The comparison procedure is performed using several synthetic *fractional Gaussian noise* and *fractional differencing noise* time series with known *Hurst* exponent. Finally, the use of well-known real LAN traffic is


Advances in Computer Science and Engineering
Research in Computing Science 27, 2007, pp. 217-227

Received 23/02/07
Accepted 08/04/07
Final version 20/04/07

used. The organization of the paper is as follows. Section 2 discusses the measurement methodology in computer network traffic studies and the mathematical theory behind *self-similar* and *long-memory* processes. Section 3. provides description of the aggregated variance method for *Hurst index* estimation and comments on the sources of inaccuracies. Section 4. presents *Variance analyzer* characteristics and functionality. The comparison procedure and results is presented in section 5. Finally, section 6 concludes the paper.

## 2 Measurement Methodology and Mathematical Theory

In order to study computer network traffic's characteristics and its effect on network performance, some measurement of real traffic must be done. The result of the measurement is a trace from which a mathematical model is fitted. Usually, there is a relationship between network performance and some parameter of the selected mathematical model, therefore, accurate parameter estimation is important. The paper assumes that the trace fitted model is either *self-similarity* or *long-memory*. Next, we describe the measurement procedure in network performance studies and the mathematical models used in the paper. In the next section we cover parameter estimation of the assumed models.

### 2.1 Measurement Methodology

The first step in a network performance study is to obtain a trace from a measurement point. The measurement point could be a point in a LAN, WAN, link and path. A well known trace could also be generated and its behavior in a link or path could be an indicator of computer network performance. Note that in general, there are several ways to obtain a representative trace that can be used for network performance studies. The paper concentrates on the study of a trace representing the number of bytes/packets/bits per time unit on a measurement point. The trace, usually contains several types of traffic coming from different sources and with different *quality of service* characteristics. This trace is sometimes called the aggregate traffic trace or the traffic rate process and can be obtained from any computer network. Formally, let $X_i$ represent the number of bytes/packets/bits for time period $(\tau_i - \tau_{i+1})$, then the trace $X = (X_i, i \in \mathbb{Z}+)$ contains the number of bytes/packes/bits for time periods $\{(\tau_i - \tau_{i+1})\}_{i \in \mathbb{Z}+}$. Note that the trace $X$ represents a discrete-time stochastic time series that can be analyzed by probabilistic means. It has been shown that *self-similar* and *long-memory* processes model well the behaviour of the traffic rate process [1] [2] [4]. Description of *self-similar* and *long-memory* processes including its relationship is described next.

### 2.2 Self-similarity

Intuitively, *self-similarity* means that the properties(e.g. correlation structure, density) of an object(e.g. a time series) are mantained independently of scaling in

time and/or space. For traffic modelling purposes, the interest is in discrete-time statistical second-order *self-similar* processes with some form of stationarity. A discrete-time stationary stochastic process, $X = (X_t, t \in \mathbb{Z}+)$, is said to be second-order *self-similar*, with *self-similarity* parameter $H = 1 - \beta/2$, called the *Hurst* parameter, if its autocorrelation function $\rho(k), k \geq 1$ follows

$$\rho(k) = \frac{1}{2}((k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta}) = g(k), \tag{1}$$

where $\beta \in (0,1)$ and $H \in (1/2, 1)$. Equation (1), implying same correlation structure in all time scales, is too strict to model network traffic, therefore, an asymptotic behaviour giving rise to equation (1) is mostly used. Let $X_a = (X^{(m)}(k), k \geq 1)$ be the aggregated process of level $m$, obtained by appliying $X^{(m)}(k) = m^{-1} \sum_{t=(k-1)m+1}^{km} X(t)$ to the original time series $X = (X_t, t \in \mathbb{Z}+)$, then a discrete-time process is said to be asymptotic second-order *self-similar* if the aggregated process' autocorrelation function behaves asymptotically as

$$\rho^{(m)}(k) \sim \frac{1}{2}((k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta}), \tag{2}$$

as $m \to \infty$, $\beta \in (0,1)$ and $H \in (1/2, 1)$. Equation (1) implies that $\rho^m(k) = g(k), \forall m \geq 1$ and equation (2) implies that $\rho^m(k) = g(k)$ only aymptotically.

### 2.3 Long-memory

Intuitively long-memory or long-range dependence means that correlations between distant points in time of a series $X_t$ are non-zero. This point relationship can occur only when the autocorrelation function behaves hyperbolically as opossed to exponentially. Long-range dependence in a stationary stochastic process, $\{X_t\}_{t \in Z_+}$ occurs when the lag $k$ autocorrelation function in $X_t$, $\rho(k) : k \geq 1$ satisfies the following asymptotic behaviour

$$\rho(k) \sim c_p k^{-\beta}, \tag{3}$$

where $c_p > 0$ is a constant and $0 < \beta < 1$. Equation (3) implies that the sum of the autocorrelation function of $X_t$ is not bounded, i.e., $\sum_{k=0}^{\infty} \rho(k) = \infty$ and the spectral density having a pole at zero, i.e., $f(\lambda) \sim c_f \mid \lambda \mid^{\beta-1}$ as $\lambda \to 0$. Another interpretation is that the correlations of a $LRD$ process decay slowly in time, thus, giving rise to non-summability of the correlations.

### 2.4 Self-similarity and Long-memory Relationship

Self-similarity and long-range dependence are closely related concepts. An asymptotic self-similar process is defined according to equation (2), now let $k \to \infty$, then

$$\rho^{(m)}(k) \sim H(2H-1)k^{-\beta}. \tag{4}$$

Equation (4) implies that in the limit(as $k \rightarrow \infty$), an asymptotic *self-similar* process is *long-range dependent*. Similarly a *long-range dependent* process $X_t$ can be constructed by the increment process of a *self-similar* process, i.e., $X_t = (Y_t - Y_{t-1}, t = 1, 2, ..)$, e.g., *fractional Gaussian noise* is obtained from the increment of a *fractional Brownian motion process*. For more information on *self-similar* and *long-range dependent* processes refer to [16] [8] [9] [15].

## 3 Parameter Estimation using Aggregated Variance

Once the mathematical model has been fitted to the measured computer network trace, estimation of some parameters characterizing the model is accomplished. Parameter estimation is important due to the relationship between parameter value and computer network performance [6] [9]. This relationship can be used for estimating computer network performance given some parameter value and for control algorithms' design in computer network performance applications [6]. Parameter estimation for *self-similar* and *long-memory* process is reduced to *Hurst-index* estimation which caracterizes completely their behaviour. Parameter estimation for these models can be done either in time-domain or frecuency-domain [12] [13] [14]. The paper concentrates on the time-domain algorithm named aggregated variance which is described next.

### 3.1 Aggregated Variance Method

Consider the aggregated series $X_a = (X^{(m)}(k), k \geq 1)$, obtained by dividing the original length $N$ series in blocks of size $m$ and computing the sample mean to each block, we take the sample variance to this series and obtain

$$Var(X^{(m)}) = \frac{1}{N/m} \sum_{k=1}^{N/m} (X_k^{(m)} - \bar{X})^2, \tag{5}$$

where $\bar{X}$ represents the original series sample mean. Equation (5) represents aggregated process' variance, mostly referred to as the aggregated variance. The aggregated variance method is based on the asymptotic behavior of the sample mean's variance in a discrete-time *self-similar* process $X_t, t \in \mathbb{Z}+$. The sample mean can be seen as the aggregated process of a discrete time series $X_t$, i.e., $E\{X_t\} = X^{(m)}$. Note that the aggregated series corresponds to the measured traffic rate process. Sample mean's variance decay, i.e., $Var(E\{X_t\})$, in *self-similar* processes behaves asymptotically as

$$Var(X^{(m)}) = Var(E\{X_t\}) \sim m^{2H-2}, \tag{6}$$

where $X_k^{(m)}$ is the aggregated process and $H$ the *Hurst-index*. Note from this result that plotting the variance of the aggregated process $X^{(m)}$ versus $m$ in log-log axes, for varying aggregation levels $m$, should result in a straight line with slope $2H - 2$. A least squares fit to the points in the plot should give the slope. Once the slope is estimated, the *Hurst-index* can easily be obtained [13].

### 3.2 Sources of Inaccuracies

Note that equation (6) gives an asymptotic behavior of the sample mean's variance, then, certain inaccuracies may appear due to the time series selected length. The longer the series the better the estimate. Additionally, due to the least squares estimate of the slope, the selection of the low and high end values of $m$, named cut-offs, affect the *Hurst-index*'s accuracy. A correct selection of these parameters(length and cut-offs)in any algorithm is then neccesary.

### 3.3 Accurate Estimation of the *Hurst-index*

Accurate estimations, as mentioned above, are obtained by the correct selection of the cut-offs and time series' length. Cut-offs and length selection, named tuning in this paper, is accomplished by first selecting the correct cut-offs and finally, based on the correct cut-offs, obtain a representative time series length. Cut-off selection is accomplished in two steps. First step is to select the high-end cut-off and the second involves determining the low-end cut-off value. High-end cut-off selection for an $H - index$ time series is accomplished by first selecting a fixed point in the $x$-axis, $x_i$, which is near the low-end regression value, then varying the points near the high-end regression value, $x_j$, $x_j \in (x_1, x_2)$. The high-end regression value which approximates better to the $H$ value is selected as the high-end cut-off. Low-end cut-off selection is accomplished by varying the low-end regression values while maintaining the selected high-end cut-off fixed, as before, the low-end regression value which approximates better to the $H$ value is selected as the low-end cut-off. Time series length selection is obtained via a cumulative analysis on the series. Cumulative analysis on a length $N$ time series, $X_t$, $t = \{1, 2, \ldots N\}$, is obtained by first dividing the original length $N$ time series in blocks of size $K \leq 1024$ and then estimating the *Hurst-index* for the series $\{X_i\}_{i=1}^{jK}$), $j = 1, 2, \ldots N/K$, i.e., $\hat{H_{jK}} = \Gamma(\{X_i\}_{i=1}^{jK})$, $j = 1, 2, \ldots N/K$, where $\Gamma(.)$ represents a *Hurst-index* estimation method. A plot of the estimated *Hurst-index* values $\hat{H_{jK}}$ versus $j$ shows the behavior of the estimated *Hurst-index*. Usually, a stability region in the plot, is an indicator of the time series required length. The cut-off and time series' length selection procedures are performed by using synthetic long-memory traces, i.e., traces with well known *Hurst-index* values.

## 4 Variance Analyzer: A Tool for Long-memory

This section presents *Variance Analyzer* main features and functionality. It also describes briefly *Selfis*, a similar tool for *long-memory* analysis.

### 4.1 Variance Analyzer

*Variance Analyzer* is a novel C++ based software tool which estimates the *Hurst-index* using the aggregated variance method. The $m$ values in the aggregated variance method in *Variance Analyzer* vary according to $10^x$, $x =$

$0.1, 0.2, \ldots, 0.1(log_{10}(N))$, where $N$ is the time series length. The selection of these values provides *Variance Analyzer* with better resolution than existing tools. An advantage of this is better accuracy but less convergence to long time series. The low-end and high-end cut-offs values are set to $10^{0.2}$ and $10^{2.2}$. These values were obtained using *fractional Gaussian noise* and FARIMA$(0, d, 0)$ synthetic long-memory traces. The time series length in these traces is set to 65536 points. Time series' *Hurst-index* estimation in *Variance Analyzer* is performed in two steps. First step involves the selection and automatic plotting of the text file. A requirement in this step is that the file should be in one-column format without spaces and comments. The non-conformance to this requirement causes *Variance Analyzer* to produce an error message. Once the file is plotted, a selection of a new file for analysis is possible. The second step involves estimating the *Hurst-index* of the selected and plotted time series. In this step, a plot of the regression points in the aggregated variance method is provided. Once the estimation is performed, *Variance Analyzer* provides functionality to return to step 1, i.e., to the orginial time series plot. As in the first step, a new file for analysis could be open. *Variance Analyzer* functionality is shown in the Petri net model of Figure 1. State **p1** is the initial state where *Variance Analyzer* is opened. State **p2** is the file open state, **p3** is the file plotted state and **p4** is the regression and *Hurst-index* estimation state. **T2**, **T5** and **T8** represent a non-valid file event and **T1**, **T4**, and **T7** represent a valid file event. Event **T3** is the plotting file event, **T6** corresponds to the estimation and regression event and **T9** represents the return event, i.e., the return to the original time series plot.

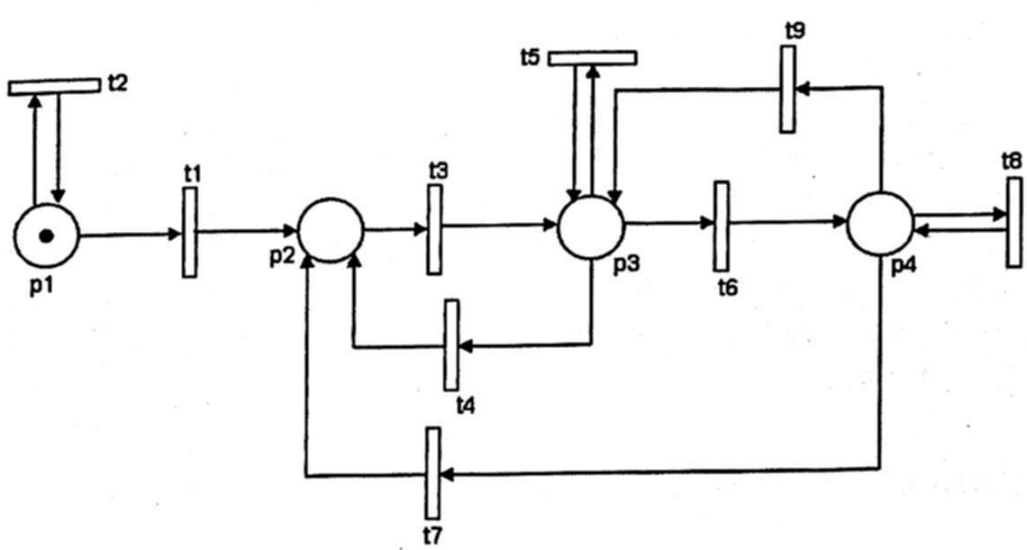


**Fig. 1.** *Variance Analyzer* Petri Net model

Figure 2 shows the user interface of *Variance Analyzer*. *Variance Analyzer* GUI consists of three main parts; the menu, the toolbar and the plotting area. The menu provides the user complete access to the functionality of *Variance*

*Analyzer* such as; file opening, program exit, *Hurst-index* estimation, etc. The toolbar provides the most often used functions such as file opening, *Hurst-index* estimation and the return button. The plotting area provides to the user the time series graphical representation and the regression points when applying the aggregated variance method.

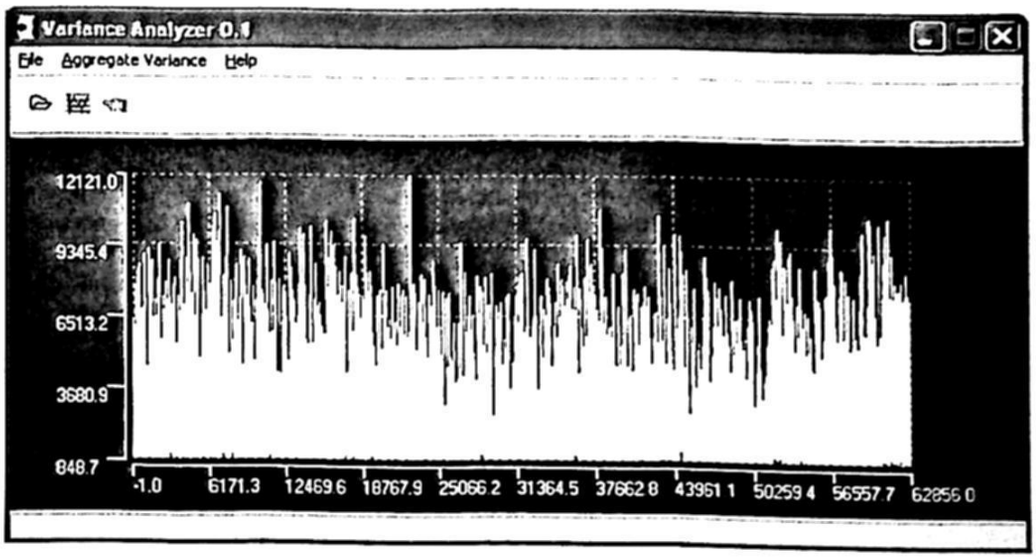


**Fig. 2.** *Variance Analyzer* User Interface

### 4.2 Selfis

Selfis is a well known software tool for *self-similarity* and *long-memory* analysis [10] [11]. *Selfis* is a Java-based software tool which estimates the *Hurst-index* using four time-domain methods and three frecuency-domain methods. It was noted that *Selfis* does not present the cut-offs used in the *Hurst-index* estimation. This paper is interested in the accuracy of *Selfis* aggregated variance method. *Selfis* aggregated variance implementation differs from that of *Variance Analyzer*, thus, our aim is to quantify the accuracy of each in estimating the *Hurst-index* for different synthetic and real *long-memory* traces. For more information on *Selfis* characteristics and functionality refer to [10] [11]

## 5 Comparison Procedure and Results

The comparison procedure of *Variance Analyzer* versus *Selfis* is presented in this section. Synthetic and real trace description is presented first, the comparison procedure is described next and finally, *Hurst-index* estimation using both tools is performed.

### 5.1 Long-memory Traces

In order to quantify the accuracy degree in *long-memory* analysis software tools, the use of traces with known *Hurst-index* are used. These traces are commonly named synthetic *long-memory* time series. The paper makes use of two synthetic *long-memory* time series types, namely *fractional Gaussian noise*($fGn$) and FARIMA$(0, d, 0)$(*fractional differencing noise,* $fDn$) time series. $fGn$ time series, are long-memory time series satisfying equation (1). $fGn$ traces were created using modified Paxson's FFT algorithm with an asymptotic mean zero decay [17] [18]. Fractional differencing noise(FARIMA$(0, d, 0)$) time series, are series satisfying

$$X_i = \Delta^{-d}\epsilon_i,\ i \geq 1, \tag{7}$$

where $\epsilon_i$ are *iid* Gaussian random variables with zero-mean and $\Delta$ is the differencing operator satistying $\Delta = \epsilon_i - \epsilon_{i-1}$. The autocovariance function of this process satisfies

$$\rho(k) = C_e k^{2d-1},\ h \to \infty, \tag{8}$$

where $d \in (-1/2, 1/2)$ and $C_e = \pi^{-1}\sigma^2\Gamma(1-2d)\sin(\pi d)$. For large lags, the $ACF$ for $fGn$ and $fDn$ has the same power decay, thus, $H = d + (1/2)$. $fDn$ time series were created using $S+$ package. Finally, the application of well-known real LAN traffic traces are used. The traces are the well-studied an classical traces of [1] [2].

### 5.2 Comparison Procedure

The comparison process was performed by using the synthetic traces described above. A set of nine $fGn$ traces with *Hurst-index* from 0.55 to 0.95 in increments of 0.05 were created. An identical set of traces for $fDn$ were also created. The length of the series, both for $fGn$ and $fDn$, was set to 65536 points. For the real trace case, the use of **AUG89.MB** and **AUG89.MP** LAN time series from Bellcore were studied. The length for these series is about 360000 points representing the number of bytes(**AUG89.MB**) and packets(**AUG89.MP**) per time unit in a LAN environment.

### 5.3 Results

Table 1 shows *Hurst-index* estimations for *Variance Analyzer* and *Selfis* using $fGn$ synthetic traces. Note that *Selfis* presents high bias for traces with *Hurst-index* 0.55 and $0.70 - 0.95$. The bias($\epsilon = H_{theoretic} - \hat{H}_{estimated}$) in *Selfis* tool for these traces is $\epsilon \geq 0.035$. Unlike *Selfis*, *Variance Analyzer* presents minimum bias estimates for the *Hurst-index* in the interval $0.55 - 0.90$. Note that *Variance Analyzer* estimations are more accurate than *Selfis* for the $fGn$ case. Table 2 shows the *Hurst-index* estimations for both tools when using FARIMA$(0, d, 0)$ long-memory time series. As can be noted from the table *Selfis* presents accurate estimates only for the $fDn$ traces with *Hurst-index* 0.60 and 0.85. *Variance*

**Table 1.** *Hurst-index* estimations using $fGn$ traces

| | Selfis | Variance Analyzer |
|---|---|---|
| *Hurst-index* | | |
| 0.55 | 0.515 | 0.5526 |
| 0.60 | 0.586 | 0.5906 |
| 0.65 | 0.673 | 0.6507 |
| 0.70 | 0.594 | 0.7019 |
| 0.75 | 0.696 | 0.7391 |
| 0.80 | 0.720 | 0.7845 |
| 0.85 | 0.795 | 0.8431 |
| 0.90 | 0.804 | 0.8739 |
| 0.95 | 0.798 | 0.9088 |

*Analyzer*, unlike *Selfis*, presents accurate estimation in the $(0.55, 0.85)$ interval. From the results it is said that *Variance Analyzer* presents more accurate estimations of the *Hurst-index* than *Selfis* both for $fGn$ and $fDn$ *long-memory* synthetic traces. The attention is now turned to the analysis of real LAN computer network time series. The study of these traces is important for testing the capability of algorithms in a real environment. Figure 3 shows *Hurst-index* estimations for **AUG89.MB** and **AUG89.MP** LAN traces. As can be seen from the table, *Selfis*, presents problems for long time series and is unable to open these types of traces. **AUG89.MB** and **AUG89.MP** length is 360000 points. Unlike *Selfis*, *Variance Analyzer* is capable of opening this file and presents accurate estimations for these traces. From this study and the above, it is seen that *Variance Analyzer* presents better accuracy either for synthetic and real *long-memory* time series.

**Table 2.** *Hurst-index* estimations using $fDn$ traces

| | Selfis | Variance Analyzer |
|---|---|---|
| *Hurst-index* | | |
| 0.55 | 0.466 | 0.5499 |
| 0.60 | 0.584 | 0.5890 |
| 0.65 | 0.570 | 0.6341 |
| 0.70 | 0.665 | 0.6921 |
| 0.75 | 0.648 | 0.7339 |
| 0.80 | 0.765 | 0.7756 |
| 0.85 | 0.841 | 0.8438 |
| 0.90 | 0.730 | 0.8674 |
| 0.95 | 0.839 | 0.9050 |

**Table 3.** *Hurst-index* estimations using real LAN traces

| | Selfis | Variance Analyzer |
|---|---|---|
| Trace(*Hurst-index*) | | |
| **AUG89.MB**($\sim 0.80$) | *NotOpened* | 0.8166 |
| **AUG89.MP**($\sim 0.90$) | *NotOpened* | 0.8662 |

## 6 Conclusions and Future Work

A tool for *long-memory* and *self-similarity* analysis for computer network time series was presented. The *Hurst-index* estimation tool, named *Variance Analyzer*, is based on the aggregated variance algorithm with tuned cut-offs. The sources of inaccuracies in the aggregated variance algorithm were identified and the correct selection of the low and high end cut-offs was proposed. A comparison procedure of *Variance Analyzer* versus *Selfis* showed that *Variance Analyzer* presents better accuracy and minimum-bias estimates of the *Hurst-index*. The comparison was performed by using known *Hurst-index* and real LAN time series. *Variance Analyzer* robustness to long time series was also accomplished. Based on this, *Variance Analyzer* should be the tool of choice when analyzing time series via the aggregated variance method for $fGn$ and $fDn$-like time series. *Variance Analyzer* could also be employed for the analysis of other non-computer network time series, e.g., geological, hydrological, etc.

## References

1. Leland, W. E., Taqqu, M.S., Willinger, W., Wilson, D. V.: On the self-similar nature of Ethernet traffic. Proc. ACM SIGCOMM '93, San Francisco CA. (1993) 183–193
2. Leland, W. E., Taqqu, M.S., Willinger, W., Wilson, D. V.: On the self-similar nature of Ethernet traffic(Extended version). IEEE/ACM Transactions on Networking **2** (1994) 1–15
3. Beran, J., Sherman, R., Taqqu, M. S., Willinger, W.: Long-range dependence in variable-bit-rate video traffic. IEEE Transactions on communications **43** (1995) 1566–1579
4. Crovella, M., Bestavros, A.: Self-similarity in Word-Wide-Web traffic: evidence and possible causes. IEEE/ACM Transactions on Networking. **5** (1997) 835–846
5. Paxson, V., Floyd, S.: Wide-area traffic: The failure of poisson modelling. IEEE/ACM Transactions on Networking. **3** (1995) 226–244
6. Park, K.: On the effect of traffic self-similarity on network performance. Proc. SPIE International conference on performance and control of network systems. (1997) 296—310
7. Taqqu, M. S., Teverovsky, V., Willinger, W.: Estimators for long-range dependence: An empirical study. Fractals. **3** (1995) 785–798
8. Beran, J.: Statistics for Long-memory Processes. New York, Chapman & Hall (1994)

9. Park, K., Willinger W.: Self-similar Network Traffic and Performance Evaluation. Wiley-Interscience (2000)
10. Karagiannis, T., Faloutsos, M., Molle, M.: A User-Friendly Self-similarity Analysis Tool. Special section on Tools and Technologies for Networking Research and Education, ACM SIGCOMM Computer Communication Review. (2003)
11. Karagiannis, T., Faloutsos, M.: SELFIS: A Tool for Self-similarity and Long-range Dependence Analysis. 1st Workshop on Fractals and Self-similarity in Data Mining: Issues and Approaches.(2002)
12. Taqqu, M. S., Teverovsky, V.: Semi-parametric graphical estimation techniques for long-memory data Athens Conference on Applied Probability and Time series analysis. **115** (1996) 420–432
13. Adler, R., Feldman, R., Taqqu, M. S.: A Practical guide To Heavy-Tails: Statistical Techniques and Applications. Boston, Birkhauser. (1998)
14. Kokoszka, P., Taqqu, M. S.: Parameter Estimation for infinite variance fractional ARIMA. Annals of Statistics. **24** (1996) 1880–1913
15. Lopez-Ardao, J., Lopez-Garcia, C., Suarez-Gonzalez, A., Fernandez-Veiga, M., Rodriguez-Rubio, R.: On the use of Self-similar Processes for Network Simulation. ACM Transactions on Modelling and Computer simulation. **10** (2000) 125–151
16. Tsybakov, B., Georganas, N.: Self-similar Processes in Communications Networks. IEEE Transactions on Infotmation Theory. **44** (1998) 1713–1725
17. Paxson, V.: Fast, Approximate Synthesis of Fractional Gaussian Noise For Generating Self-similar Network Traffic. Computer Communications Review. **27** (1997) 5–18
18. Rolls, D.: Improved Fast Approximate Synthesis of Fractional Gaussian Noise. Hawaii International Conference on Statistics and Related Fields. (2002)