FGN과 Daubechies Wavelets을 이용한 빠른 Self-Similar 네트워크 Traffic의 생성

정 해 덕[†] · 이 종 숙[††]

요 약

최근의 통신 네트워크에서 teletraffic의 양상은 Poisson 프로세스보다 self-similar 프로세스에 의해서 더 잘 반영된다. 이는 통신 네트워크의 teletraffic에 관련하여 self-similar한 성질을 고려하지 않는다면, 통신 네트워크의 성능에 관한 결과는 부정확 할 수밖에 없다는 의미가 된다. 따라서, 통신 네트워크에 관한 시뮬레이션을 수행하기 위한 매우 중요한 요소 중에 하나는 충분히 긴 self-similar한 sequence를 얼마나 잘 생성하느냐의 문제이다. 본 논문에서는 fractional Gaussian noise와 wavelet 변환을 이용한 새로운 pseudo-random self-similar sequence 생성기를 구현 및 분석하였다. 특별히 본 생성기는 다른 wavelet 변환보다 long range dependent한 프로세스들의 self-similar 구조에 잘 맞기 때문에 좀더 정확한 결과를 유도할 수 있는 Daubechies wavelet을 사용하였다. 본 생성기를 이용하여 매우 긴 sequence를 생성하는데 요구되는 통계적인 정확도와 생성시간에 대해서 분석하였으며, 본 논문에서 제안한 생성기의 성능은 Hurst 변수의 상대적인 정확도로 보았을 때, 그리고 sequence의 생성시간을 고려했을 때에 매우 우수함을 보였다. 이 생성기의 이론적 complexity는 n개의 난수를 발생하는데 $O(n)$이 요구된다.

Fast Self-Similar Network Traffic Generation Based on FGN and Daubechies Wavelets

HaeDuck J. Jeong[†] · JongSuk R. Lee[††]

ABSTRACT

Recent measurement studies of real teletraffic data in modern telecommunication networks have shown that *self-similar* (or *fractal*) processes may provide better models of teletraffic in modern telecommunication networks than Poisson processes. If this is not taken into account, it can lead to inaccurate conclusions about performance of telecommunication networks. Thus, an important requirement for conducting simulation studies of telecommunication networks is the ability to generate long synthetic stochastic self-similar sequences. A new generator of pseudo-random self-similar sequences, based on the fractional Gaussian nois and a wavelet transform, is proposed and analysed in this paper. Specifically, this generator uses Daubechies wavelets. The motivation behind this selection of wavelets is that Daubechies wavelets lead to more accurate results by better matching the self-similar structure of long range dependent processes, than other types of wavelets. The statistical accuracy and time required to produce sequences of a given (long) length are experimentally studied. This generator shows a high level of accuracy of the output data (in the sense of the Hurst parameter) and is fast. Its theoretical algorithmic complexity is $O(n)$.

키워드 : Self-similar 프로세스(Self-Similar Process), Teletraffic 생성기(Teletraffic Generator), Fractional Gaussian Noise, Daubechies Wavelet, Complexity, Hurst 변수(Hurst Parameter)

1. Introduction

The search for accurate mathematical models of data streams in modern telecommunication networks has attracted a considerable amount of interest in the last few years. Several recent teletraffic studies of local and wide area networks, including the World Wide Web, have shown that commonly used teletraffic models, based on Poisson or related processes, are not able to capture the *self-similar* (or *fractal*) nature of teletraffic [1-4], especially when these networks are engaged in such sophisticated services as variable-bit-rate (VBR) video transmission [5-7]. The properties of teletraffic in such scenarios are very different from both the properties of conventional models of telephone traffic and the traditional models of data traffic generated by computers.

The use of traditional models of teletraffic can result in overly optimistic estimates of performance of telecommunication networks, insufficient allocation of communication

and data processing resources, and difficulties in ensuring the quality of service expected by network users [3, 8, 9]. On the other hand, if the strongly correlated character of teletraffic is explicitly taken into account, this can also lead to more efficient traffic control mechanisms.

Several methods for generating pseudo-random self-similar sequences have been proposed. They include methods based on fast fractional Gaussian noise [10], fractional ARIMA processes [11], the $M/G/\infty$ queue model [1, 6], autoregressive processes [12], spatial renewal processes [13], etc. Some of them generate asymptotically self-similar sequences and require large amounts of CPU time. For example, Hosking's method [11], based on the F-ARIMA(0, d, 0) process, needs 1.5 hours to produce a self-similar sequence with 131,072 (2^{17}) numbers on a Pentium II [1, 14]. It requires $O(n^2)$ computations to generate n numbers. Even though exact methods of generation of self-similar sequences exist (for example [10]), they are only fast enough for short sequences. They are usually inappropriate for generating long sequences because they require multiple passes along generated sequences. To overcome this, approximate methods for generation of self-similar sequences in simulation studies of telecommunication networks have been proposed [15, 16].

The evaluation of our generator based on Daubechies wavelets (DW) concentrates on two aspects. (i) how accurately a self-similar process can be generated ; and (ii) how quickly the method generates long self-similar sequences. Our method, based on the fractional Gaussian noise (FGN) and Daubechies wavelets, will be called the FGN-DW method.

A summary of the basic properties of self-similar processes is given in Section 2. Section 3 describes the spectral density of FGN processes, while a discrete wavelet transform (DWT) for synthesising approximate FGN is presented in Section 4. In Section 5, a generator of pseudo-random self-similar sequences, based on FGN and DW, is described. Numerical results of analysis of sequences generated by this generator are discussed in Section 6.

2. Self-Similar Processes and Their Properties

2.1 Mathematical Definition of Self-Similarity

One can distinguish two types of stochastic self-similarity. A continuous-time stochastic process Y_t is strictly *self-similar* with a self-similarity parameter $H(1/2 < H < 1)$,

if Y_{ct} and $c^H Y_t$ (the rescaled process with time scale ct) have identical finite-dimensional probability for any positive time stretching factor c [17-19]. This means that, for any sequence of time points $t_1, t_2, \cdots, t_n$, and for any $c > 0$,

$$\{Y_{ct_1}, Y_{ct_2}, \cdots, Y_{ct_n}\} \doteq \{c^H Y_{t_1}, c^H Y_{t_2}, \cdots, c^H Y_{t_n}\},$$

where $\doteq$ denotes equivalence in distribution. This definition of the strictly self-similarity is in a sense of probability distribution (or narrow sense), quite different from that of the second-order self-similar process (or self-similar process in a broad sense). Self-similarity in the broad sense is observed at the mean, variance and auto-correlation level, whereas self-similarity in the narrow sense is observed at the probability distribution level.

When the weakly continuous-time self-similar process Y_t has stationary increments, i.e., the finite- dimensional probability distributions of $Y_{t_0 + t} - Y_{t_0}$ do not depend on t_0, we can construct a stationary incremental process $X = \{X_i = Y_{i+1} - Y_i : i = 0, 1, 2, \cdots\}$. Namely, in the discrete-time case, let X be a (discrete-time) stationary incremental process with mean $\mu = E[X]$, variance $\sigma^2 = E[(X - \mu)^2]$, and (normalised) auto-correlation function (ACF) $\{\rho_k\}$, $k = 0, 1, 2, \cdots$, where

$$\rho_k = \frac{E[(X_i - \mu)(X_{i+k} - \mu)]}{\sigma^2}. \tag{1}$$

X is strictly stationary if $\{X_{i_1}, X_{i_2}, \cdots, X_{i_n}\}$ and $\{X_{i_1 + k}, X_{i_2 + k}, \cdots, X_{i_n + k}\}$ ossess the same joint distribution. However, we limit our attention to processes with a weaker form of stationarity, i.e., second-order stationarity (or weak, broad, or wide sense stationarity). Let $X^{(m)} = \{X_1^{(m)}, X_2^{(m)}, \cdots\}$, $m = 1, 2, 3, \cdots$, be a sequence of batch means, that is,

$$X_i^{(m)} = \frac{1}{m}(X_{im - m + 1} + \cdots + X_{im}), \ i \geq 1, \tag{2}$$

and let $\{\rho_k^{(m)}\}$ denote the ACF of $X^{(m)}$. The process X is called *exactly second-order self-similar* with $1/2 < H < 1$, if for all $m \geq 1$,

$$\rho_k^{(m)} = \rho_k, \ k \geq 0. \tag{3}$$

In other words, the process X and the aggregated processes $X^{(m)}$, $m \geq 1$, have an identical correlation structure.

The process X is *asymptotically second-order self-similar* with $1/2 < H < 1$, if for all k large enough,

$$\rho_k^{(m)} \rightarrow \rho_k, \text{ as } m \rightarrow \infty. \tag{4}$$

The most frequently studied models of self-similar traffic belong either to the class of fractional autoregressive integrated moving-average (F-ARIMA) processes or to the class of fractional Gaussian noise processes ; see [1, 11, 16]. F-ARIMA(p, d, q) processes were introduced by Hosking [11] who showed that they are asymptotically self-similar with Hurst parameter $H = d + 1/2$, as long as $0 < d < 1/2$. where p is the order of autoregression in the ARIMA process and q is the order of the moving average in the ARIMA process. For the second class, the FGN process is the incremental process $\{Y_k\} = \{X_k - X_{k-1}\}$, $k \geq 0$, where $\{X_k\}$ designates a fractional Brownian motion (FBM) random process. This process is a (discrete-time) stationary Gaussian process with mean μ, variance σ^2 and $\{\rho_k\} = \{1/2$ $(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H})\}$, $k > 0$. An FBM process, which is the sum of FGN increments, is characterized by three properties [20]. (i) it is a continuous zero-mean Gaussian process $\{X_t\} = \{X_s : s \geq 0 \text{ and } 0 < H < 1\}$ with ACF given by $\rho_{s,t} = 1/2(s^{2H} + t^{2H} - |s-t|^{2H})$ where s is time lag and t is time ; (ii) its increments $\{X_t - X_{t-1}\}$ form a stationary random process ; (iii) it is self-similar with Hurst parameter H, that is, for all $c > 0$, $\{X_{ct}\} \approx \{c^H X_t\}$, in the sense that, if time is changed by the ratio c, then $\{X_t\}$ is changed by c^H.

2.2 Properties of Long-Range Dependent Self-Similar Processes

Main properties of self-similar processes include ([1, 17, 21]).

• *Long-range dependence* : A process $\{X_k\}$ is called a stationary process with *long-range dependence* (LRD) if its ACF $\{\rho_k\}$ is non-summable, i.e., $\sum_{k=0}^{\infty} \rho_k = \infty$. The speed of decay of auto-correlations is more hyperbolic than exponential. Another definition of LRD is given by

$$\rho_k \sim L(t) k^{-(2-2H)}, \text{ as } k \rightarrow \infty, \tag{5}$$

where $1/2 < H < 1$ and $L(\cdot)$ slowly varies at infinity, i.e., $\lim_{t \to \infty} \frac{L(xt)}{L(t)} = 1$, for all $x > 0$; see [1]. The Hurst

parameter H characterises the relation in (5), which specifies the form of the tail of the ACF.

• *Slowly decaying variance* : The variance of the sample mean decreases more slowly than the reciprocal of the non-overlapping batch size m, i.e., $Var[\{X_k^{(m)}\}] \rightarrow c_1 m^{-\beta_1}$, as $m \rightarrow \infty$, where c_1 is a constant and $0 < \beta_1 < 1$.

• *Hurst effect* : Historically, the importance of self-similar processes lies in the fact that they provide an elegant explanation and interpretation of strong correlations in some empirical data. Namely, for a given sequence of random variables $X = \{X_i\}_{i=1}^{n} = \{X_1, X_2, \cdots, X_n\}$, one can consider the so-called *rescaled adjusted range* $R(t, m) / S(t, m)$ (or R/S-statistic), with

$$R(t, m) = \max_i [N_{t+i} - N_t - \frac{i}{m}(N_{t+m} - N_t), 0 \leq i \leq m]$$
$$- \min_i [N_{t+i} - N_t - \frac{i}{m}(N_{t+m} - N_t), 0 \leq i \leq m], \tag{6}$$

where $1 \leq t \leq n$, m is the batch size and $N_t = \sum_{i=1}^{t} X_i$; and

$$S(t, m) = \sqrt{m^{-1} \sum_{i=t+1}^{t+m} (X_i - \overline{X}_{t,m})^2}, \tag{7}$$

where $\overline{X}_{t,m} = m^{-1} \sum_{i=t+1}^{t+m} X_i$.

Hurst found empirically that for many time series observed in nature, the expected value of $R(t, m)/S(t, m)$ asymptotically satisfies the power-law relation :

$$E[\frac{R(t, m)}{S(t, m)}] \rightarrow c_2 m^H, \text{ as } m \rightarrow \infty, \text{ with } 1/2 < H < 1,$$

where c_2 is a finite positive constant [17]. This empirical finding was in contradiction to previously known results for Markovian and related processes. For a stationary process with SRD, $E[R(t, m)/S(t, m)]$ behaves asymptotically like a constant times $m^{1/2}$. Therefore, for large values of m, the R/S-statistic plot is randomly scattered around a straight line with slope $1/2$. Hurst's finding that for the Nile River data, and for many other hydrological, geophysical, and climatological data, $R(t, m)/S(t, m)$ is randomly scattered around a straight line with slope $H > 1/2$, is known as the *Hurst effect*, and H is known as the Hurst parameter (or self-similarity parameter). Mandelbrot and Wallis [20] showed that the Hurst effect can be modelled by FGN with the self-sim-

ilarity parameter $1/2 < H < 1$.

- *1/f-noise* : The spectral density $f(\lambda ; H)$ obeys a power law near the origin, i.e., $f(\lambda ; H) \to c_3\lambda^{1-2H}$, as $\lambda \to 0$, where c_3 is a finite positive constant and $1/2 < H < 1$.

We will use these properties to investigate characteristics of generated self-similar sequences.

3. Spectral Density of FGN Processes

In our generator, numbers representing the spectral density function of FGN are obtained by applying appropriate transformations to originally uniformly distributed pseudo-random numbers. The spectral density $f(\lambda ; H)$ of an FGN process is given by

$$f(\lambda ; H) = 2c_f(1 - \cos(\lambda)) \sum_{k=-\infty}^{\infty} |2\pi k + \lambda|^{-2H-1} \quad (8)$$

for $0 < H < 1$ and $-\pi \le \lambda \le \pi$, where $c_f = \sigma^2(2\pi)^{-1}\sin(\pi H)\Gamma(2H+1)$ and $\sigma^2 = Var[X_k]$; see [17].

The main difficulty with using Equation (8) to compute the spectral density is the vexing infinite summation. The approximation of the above $f(\lambda ; H)$ is given in [17] as

$$f(\lambda ; H) = c_f|\lambda|^{1-2H} + O(|\lambda|^{min(3-2H,2)}) \quad (9)$$

where $O(\cdot)$ represents the residual error.

A generator of self-similar sequences based on FGN was also proposed by Paxson [16], but his method was based on a more complicated approximation of $f(\lambda ; H)$ than this one given by Equation (9).

4. Discrete Wavelet Transform

Our method for generating synthetic self-similar FGN sequences in a time domain is based on the discrete wavelet transform (DWT). It has been shown that wavelets can provide compact representations for a class of FGN processes [22, 23]. This is because the structure of wavelets naturally matches the self-similar structure of the long range dependent processes [24]. Wavelets are complete orthonormal bases which can be used to represent a random time series in two domains : time and frequency. In Hilbert space $L^2(R)$, scaled and shifted functions $\psi_{j,m}(k)$ of wavelets can be represented as $\psi_{j,m}(k) = 2^{-j/2}\psi_0(2^{-j}k - m)$ where j and m are positive integers [25]. Since such wave-

lets are obtained by scaling and shifting a single function, $\psi_0(k), \psi_0(k)$ is called the mother wavelet. Moreover, all base functions $\psi_{j,m}(k)$ have the same shape as the mother wavelet and therefore are self-similar with each other.

For our generator, we chose Daubechies wavelets, which belong to the class of orthonormal wavelets, because they produce more accurate coefficients of wavelets than Haar wavelets (for more detailed discussions, see also literature in [26, 27] ; and our results of the comparison in Section 6). They are defined as $\phi(k) = \sum_{i=-2^S+1}^{1} (-1)^i p_{1-i}\phi(2k-i)$, where $\{p_i\}$ is the two-scale sequence of $\phi(k)$ and $\phi(k) = \sum_{i=0}^{2^S} p_i\phi(2k-i)$. A discrete-time process $\{X_k\}$ can be represented through its inverse DWT $\{X_k\} = \sum_{j=1}^{S} \sum_{m=0}^{2^{S-j}-1} d_{j,m} \psi_{j,m}(k)$, where $0 \le k < 2^S$; and S is a positive integer which characterizes the limited resolution in time ; and $d_{j,m}$ s are wavelet coefficients which can be obtained through the DWT, since $d_{j,m} = \sum_{t=0}^{2^S-1} X_k\psi_{j,m}(k)$.

4.1 Daubechies Wavelets

Daubechies discovered one of the original wavelet families [26]. This family contains a single wavelet for each possible even valued filter length, beginning with four coefficients. As the filter length grows, the wavelets move from highly localized (due to the small number of non-zero filter coefficients) to highly smooth (for larger numbers of coefficients). The formation of the smallest of this family of wavelets, often referred to as Daub(4), which stands for the Daubechies wavelets with four coefficients, is calculated below [26].

The transformation matrix for a Daubechies wavelet of length four is given by the below matrix.

$$\begin{bmatrix} c_0 & c_1 & c_2 & c_3 & 0 & 0 & 0 & \cdots \\ c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & c_0 & c_1 & c_2 & c_3 & 0 & \cdots \\ 0 & 0 & c_3 & -c_2 & c_1 & -c_0 & 0 & \cdots \\ & & & \cdots & & & & \\ & & & & \cdots & & & \\ & & & & & \cdots & & \\ & & & & & \cdots & c_0 & c_1 & c_2 & c_3 \\ & & & & & \cdots & c_3 & -c_2 & c_1 & -c_0 \\ c_2 & c_3 & 0 & \cdots & & \cdots & 0 & 0 & c_0 & c_1 \\ c_1 & -c_0 & 0 & \cdots & & \cdots & 0 & 0 & c_3 & -c_2 \end{bmatrix} \quad (10)$$

The odd rows of this matrix represent a convolution of

the data vector with the coefficients $[c_0, c_1, c_2, c_3]$. These rows can be efficiently calculated due to its sparsity (requiring four multiplications and three additions per data value). The even rows of this matrix represent a convolution of the data vector with the coefficients $[c_3, -c_2, c_1, -c_0]$. They are also efficiently calculated, requiring the same number of operations. The total effect of this operation is to convolve the data vector with two different four-coefficient filters and reduce the results by a factor of two.

The inverse (reconstruction) transform is the inverse of this matrix, and is equal to its transpose. It can be used to generate the following requirements for the values $[c_0, c_1, c_2, c_3]$.

$$c_0^2 + c_1^2 + c_2^2 + c_3^2 = 1$$
$$c_0 \times c_2 + c_1 \times c_3 = 0.$$

For the values of these coefficients to form the desired high- and low-pass filters, we must also require :

$$c_3 - c_2 + c_1 - c_0 = 0$$
$$0 \times c_3 - 1 \times c_2 + 2 \times c_1 - 3 \times c_0 = 0.$$

The unique solution to these four equations is.

$$c_0 = (1 + \sqrt{3})/4 \times \sqrt{2}$$
$$c_1 = (3 + \sqrt{3})/4 \times \sqrt{2}$$
$$c_2 = (3 - \sqrt{3})/4 \times \sqrt{2}$$
$$c_3 = (1 - \sqrt{3})/4 \times \sqrt{2}.$$

This same general method can be used to form Daubechies wavelet filter values for any even filter length greater than four.

The Daub(4) DWT of an input data vector is calculated as follows (this method assumes a data vector length that is a power of two; the use of other lengths requires special treatment).

Given the data vector.

$$[y_0 \ y_1 \ y_2 \ y_3 \ y_4 \ y_5 \ y_6 \ y_7 \ y_8 \ y_9 \ y_{10} \ y_{11} \ y_{12} \ y_{13} \ y_{14} \ y_{15}],$$

transform it with (multiply by) the forward matrix (10) to give :

$$[y_0 \ d_0 \ y_1 \ d_1 \ y_2 \ d_2 \ y_3 \ d_3 \ y_4 \ d_4 \ y_5 \ d_5 \ y_6 \ d_6 \ y_7 \ d_7],$$

where s_* are smooth responses (low-pass) and d_* are detail responses (high-pass). This vector is then permuted

to collect the smooth and detail areas as follows :

$$[s_0 \ s_1 \ s_2 \ s_3 \ s_4 \ s_5 \ s_6 \ s_7 \ d_0 \ d_1 \ d_2 \ d_3 \ d_4 \ d_5 \ d_6 \ d_7].$$

For a DWT, this process is then iteratively repeated on the smooth values to obtain the following :

$$[S_0 \ D_0 \ S_1 \ D_1 \ S_2 \ D_2 \ S_3 \ D_3 \ d_0 \ d_1 \ d_2 \ d_3 \ d_4 \ d_5 \ d_6 \ d_7].$$

and then permuted to.

$$[S_0 \ S_1 \ S_2 \ S_3 \ D_0 \ D_1 \ D_2 \ D_3 \ d_0 \ d_1 \ d_2 \ d_3 \ d_4 \ d_5 \ d_6 \ d_7].$$

In this example we must now stop, as the length of the next smooth sub-vector (length four) equals the length of the analysis filter (length four). Reconstruction is an exact reversal of this procedure, using the transpose (inverse) of the forward transform matrix.

For our generator, we chose Daubechies wavelets because they produce more accurate coefficients of wavelets than Haar wavelets (for more detailed discussions, see also [26, 27] ; and our results of the comparison in Section 6.

5. A Fast Algorithm for Generating Self-Similar Teletraffic

We present a new generator of pseudo-random self-similar sequences based on fractional Gaussian noise (FGN) and Daubechies wavelets (DW), called the FGN-DW method [28]. A pseudo-random generator of self-similar teletraffic based on Haar wavelet transforms has been proposed in [29, 30] and [31]. We used Daubechies wavelets because the generator based on Daubechies wavelets produces more accurate self-similar sequences than one based on Haar wavelets. In other words, not only estimates of H obtained from the Daubechies wavelets are closer to the true values than those from the Haar wavelets, but also variances obtained from the Daubechies wavelets are lower. The reason behind is that the Daubechies wavelets produce smoother coefficients of wavelets that are used in the discrete wavelet transform than the Haar wavelets [26, 27, 32]. Haar wavelets are discontinuous, and they do not have good time-frequency localisation properties, since their Fourier transforms decay as $|\lambda|^{-1}$, for $\lambda \to \infty$, meaning that the resulting decomposition has a poor scale. Therefore, Daubechies wavelets produce more accurate coefficients than Haar wavelets; for a more detailed discussion, see [26, 27].

Our method for generating synthetic self-similar FGN

sequences in a time domain is based on a discrete wavelet transform (DWT). Wavelets can provide compact representations for a class of FGN processes [22, 23, 32], because the structure of wavelets naturally matches the self-similar structure of long-range dependent processes [24, 26, 27].

We claim that the FGN-DW method is sufficiently fast for the practical generation of synthetic self-similar sequences that can be used as simulation input data. The general strategy behind our method is similar to Paxson's, who used the Fourier transform [16]. Figure 1 graphically illustrates a discrete Fourier and a discrete wavelet transform. Wavelet analysis transforms a sequence onto a time-scale grid, where the term *scale* is used instead of *frequency*, because the mapping is not directly related to frequency as in the Fourier transform. The wavelet transform delivers good resolution in both time and scale, as compared to the Fourier transform, which provides only good frequency resolution. The algorithm consists of the following steps.

Step 1 : Given : H. Start for $i = 1$ and continue until $i = n$. Calculate a sequence of values $\{f_1, f_2, \cdots, f_n\}$ us-

ing Equation (9), where $f_i = \hat{f}(\pi i/n; H)$, corresponding to the spectral density of an FGN process for frequencies f_i ranging between π/n and π. The main difficulty with using Equation (8) when computing the spectral density is that it requires to execute the infinite summation.

This formula was used in the generation of self-similar sequences proposed in this paper. Another generator of self-similar sequences based on FGN was also proposed by Paxson [16], but his method was based on a more complicated approximation of $f(\lambda, H)$ as shown in Equation (8). Equation (9) can be used to determine $f(\lambda, H)$ for $\lambda \to \infty$, or for $n \to \infty$ at $\lambda = \pi/n$. For a large value of λ, $f(\lambda, H)$ can be calculated by Equation (8).

Step 2 : Multiply $\{f_i\}$ by realisations of an independent exponential random variable with a mean of one to obtain $\{\hat{f}_i\}$, because the spectral density estimated for a given frequency is distributed asymptotically as an independent exponential random variable with mean $f(\lambda, H)$ [8].
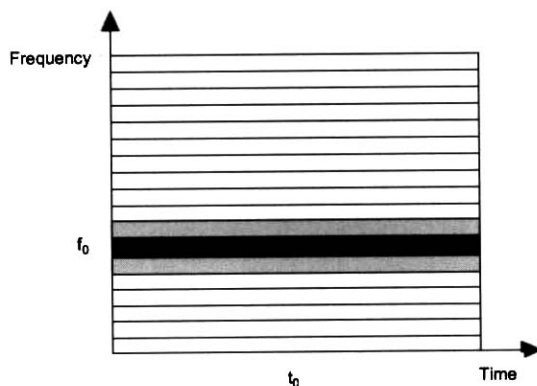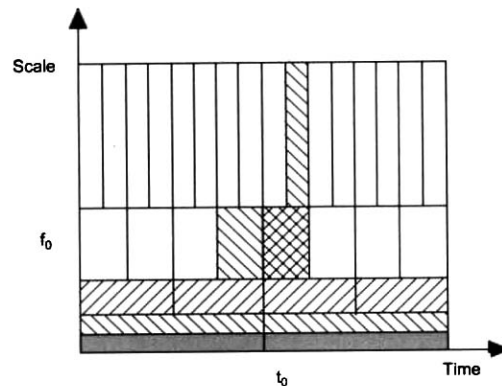
(a) A discrete Fourier transform (b) A discrete wavelet transform

(Figure 1) A graphical representation of a discrete Fourier transform and a discrete wavelet transform

Step 3 : Generate a sequence $\{Y_1, Y_2, \cdots, Y_p\}$ of complex numbers such that $|Y_i| = \sqrt{\hat{f}_i}$ and the phase of Y_i is uniformly distributed between 0 and 2π. This random phase technique, taken from Schiff [33], preserves the spectral density corresponding to $\{\hat{f}_i\}$. It also makes the marginal distribution of the final sequence normal and produces the requirements for FGN.

Step 4 : Calculate the two synthetic coefficients of ortho-

normal Daubechies wavelets that are used in the inverse DWT (IDWT) [34] ; also see Appendix A. The output sequence $\{X_1, X_2, \cdots, X_n\}$ representing approximately self-similar FGN process (in time domain) is obtained by applying the IDWT operation to the sequence $\{Y_1, Y_2, \cdots, Y_n\}$.

Using the previous steps, the proposed FGN-DW method generates a fast and sufficiently accurate self-similar FGN process $\{X_1, X_2, \cdots, X_n\}$. (Appendix A provides a program

written in Matlab for implementing this method using the pyramidal algorithm of IDWT) It took 16 seconds to generate a sequence of 1,048,576 numbers on a Pentium II (233 MHz, 512 MB). Its theoretical algorithmic complexity is $O(n)$. Moreover, the accuracy of Daubechies wavelets is slightly better than Haar wavelets, but there is no difference in the time taken to obtain the same number of coefficients.

6. Analysis of Self-Similar Sequences

The generator of self-similar sequences of self-similar pseudo-random numbers described in Section 5 has been implemented in Matlab on a Pentium II (233 MHz, 512 MB); see Appendix A. The mean times required for generating sequences of a given length were obtained by using the Matlab **clock** command and averaged over 30 iterations, having generated sequences of 32,768 (2^{15}), 131,072 (2^{17}), 262,144 (2^{18}), 524,288 (2^{19}) and 1,048,576 (2^{20}) numbers.

We have also analysed the efficiency of the method. For each of H = 0.6, 0.7, 0.8 and 0.9, the method was used to generate over 100 sample sequences of 32,768 (2^{15}) numbers starting from different random seeds. Self-similarity and marginal distributions of the generated sequences were assessed by applying the best currently available estimators : the wavelet-based H estimator and Whittle's MLE.

- *Whittle's approximate maximum likelihood estimate (MLE)* : for a more refined data analysis, used to obtain confidence intervals (CIs) for the Hurst parameter H [17]. On the other hand, it examines the properties in frequency domain, while the R/S statistic plot and variance-time plot focus on the time domain. Suppose $\{x_1, x_2, \cdots, x_n\}$ is a sequence of a self-similar process $\{X_k\}$ for which all parameters are known except $Var[X_i]$ and H. Let $f(\lambda ; H)$ be the spectral density of $\{X_k\}$ when normalised to have variance 1, and $I(\lambda)$ be the periodogram of $\{X_k\}$. Then to estimate H, find $\hat{H}$ that minimises the following equation : $g(\hat{H}) = \int_{-\pi}^{\pi} I(\lambda) / f(\lambda ; \hat{H}) d\lambda$.

- *Wavelet-based H estimator* : The original wavelet-based H estimator, proposed by Abry and Veitch in 1998 [24], is a computationally simple and fast estimator based on a wavelet transform, with a fast pyramidal algorithm for the wavelet transform, with its complexity

of order $O(n)$. However, as argued in [35] this estimator suffers from a bias associated with its log-log regression component. Later, Veitch and Abry [35] proposed a new improved method of estimation of the H parameter within a so-called wavelet-based joint estimator, which allows us to estimate both H and so-called power parameter, an independent quantitative parameter with the unit of variance; see [35] and [36] for details. The resulted wavelet-based H estimator, that we further simply call the wavelet-based H estimator, is asymptotically unbiased and (almost) the most efficient [35]. For detailed discussions, see [37].

6.1 Comparison of Daubechies Wavelets and Haar Wavelets for Generation of LRD Sequences

The estimates of the Hurst parameter obtained from the wavelet-based H estimator and Whittle's MLE, have been used to analyse the accuracy of the generator. The relative inaccuracy ΔH is calculated using the formula : $\Delta H = ((\hat{H} - H)/H) * 100\%$, where H is the required value of the Hurst parameter and $\hat{H}$ is the empirical mean value over a number of independently generated sequences. The presented numerical results are all averaged over 100 sequences.

Comparison results of sequences produced by generators based on Haar and Daubechies wavelets with several coefficients are shown in Tables 1 and 2. The relative error associated with each wavelet were also compared. The results indicate that Daubechies wavelets with sixteen coefficients produce the most accurate results and are slightly more accurate than Haar wavelets. In addition, <Table 3> and <Table 4> show that the variances obtained from Daubechies wavelets are smaller than those obtained from Haar wavelets. However, the two wavelets theoretically require the same $O(n)$ operations to transform n coefficients of their wavelets, and there is no difference in the time required to obtain a given number of their coefficients. For more detailed discussions of Haar wavelets, see [26, 27].

6.2 Analysis of Hurst parameters

The results for the wavelet-based H estimator and Whittle's MLE of H with the corresponding 95% CIs $\hat{H} \pm 1.96 \hat{\sigma}_{\hat{H}}$, see <Table 1> and <Table 2> for Daub(16), show that for all input H values, the FGN-DW method produces sequences with negatively biased H values, except

H = 0.6, for the wavelet-based H estimator. The FGN-DW   method demonstrated a high level of accuracy and was fast.

〈Table 1〉 Comparison of Daubechies wavelets and Haar wavelets : mean values of estimated H obtained using the wavelet-based H estimator for H = 0.6, 0.7, 0.8 and 0.9. Daub(♯) stands for the Daubechies wavelets with ♯ coefficients.

| Methods | Mean Values of Estimated H and ΔH | | | | | | | |
| | .6 | | .7 | | .8 | | .9 | |
	$\overline{H}$	$\Delta H(\%)$	$\overline{H}$	$\Delta H(\%)$	$\overline{H}$	$\Delta H(\%)$	$\overline{H}$	$\Delta H(\%)$
Haar	.6073 (.580, .635)	+1.220	.7141 (.687, .742)	+2.018	.8217 (.794, .849)	+2.712	.9299 (.902, .958)	+3.327
Daub(2)	.6019 (.574, .630)	+0.323	.6984 (.671, .726)	-0.229	.7943 (.767, .822)	-0.709	.8898 (.862, .917)	-1.137
Daub(4)	.6026 (.575, .630)	+0.433	.7039 (.676, .731)	+0.554	.8055 (.778, .833)	+0.684	.9074 (.880, .935)	+0.821
Daub(8)	.6026 (.575, .630)	+0.430	.7031 (.676, .731)	+0.445	.8039 (.776, .831)	+0.486	.9049 (.877, .932)	+0.545
Daub(16)	.6013 (.574, .629)	+0.214	.6987 (.671, .726)	-0.185	.7962 (.769, .824)	-0.474	.8938 (.866, .921)	-0.694

〈Table 2〉 Comparison of Daubechies wavelets and Haar wavelets : mean values of estimated H obtained using Whittle's MLE for H = 0.6, 0.7, 0.8 and 0.9. Daub(♯) stands for the Daubechies wavelets with ♯ coefficients.

| Methods | Mean Values of Estimated H and ΔH | | | | | | | |
| | .6 | | .7 | | .8 | | .9 | |
	$\overline{H}$	$\Delta H(\%)$	$\overline{H}$	$\Delta H(\%)$	$\overline{H}$	$\Delta H(\%)$	$\overline{H}$	$\Delta H(\%)$
Haar	.5766 (.567, .586)	-3.905	.6567 (.647, .666)	-6.188	.7399 (.731, .749)	-7.508	.8256 (.817, .835)	-8.263
Daub(2)	.5814 (.572, .591)	-3.106	.6662 (.657, .676)	-4.829	.7535 (.744, .763)	-5.813	.8428 (.834, .852)	-6.357
Daub(4)	.5836 (.574, .593)	-2.736	.6702 (.661, .680)	-4.254	.7590 (.750, .768)	-5.127	.8494 (.840, .859)	-5.620
Daub(8)	.5845 (.575, .594)	-2.578	.6719 (.663, .681)	-4.017	.7612 (.752, .770)	-4.854	.8520 (.843, .861)	-5.336
Daub(16)	.5849 (.575, .594)	-2.521	.6725 (.663, .682)	-3.924	.7620 (.753, .771)	-4.745	.8530 (.844, .862)	-5.223

〈Table 3〉 Variances of estimated H obtained using the wavelet-based H estimator for Daubechies wavelets and Haar wavelets for H = 0.6, 0.7, 0.8 and 0.9. Daub (♯) stands for the Daubechies wavelets with ♯ coefficients.

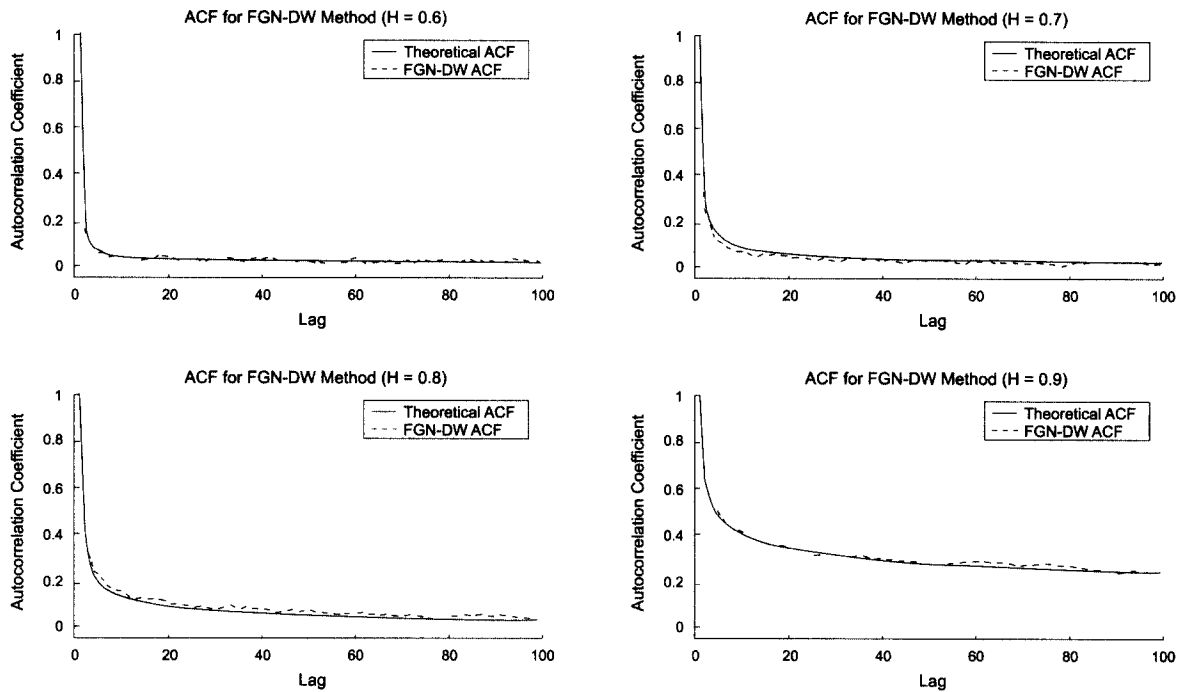| Methods | Variances of Estimated H | | | |
	.6	.7	.8	.9
Haar	2.070e-04	2.263e-04	2.534e-04	2.923e-04
Daub(2)	1.949e-04	2.154e-04	2.393e-04	2.670e-04
Daub(4)	2.508e-04	2.540e-04	2.586e-04	2.645e-04
Daub(8)	2.440e-04	2.432e-04	2.438e-04	2.456e-04
Daub(16)	2.116e-04	2.055e-04	2.007e-04	1.973e-04

6.3 Autocorrelation Test and Sequence Plot for the FGN-DW Method

ACFs characterise the correlation structure and are used to investigate the behaviour of self-similar sequences. For H = 0.6, 0.7, 0.8 and 0.9, 〈Figure 2〉 shows the ACF of se-
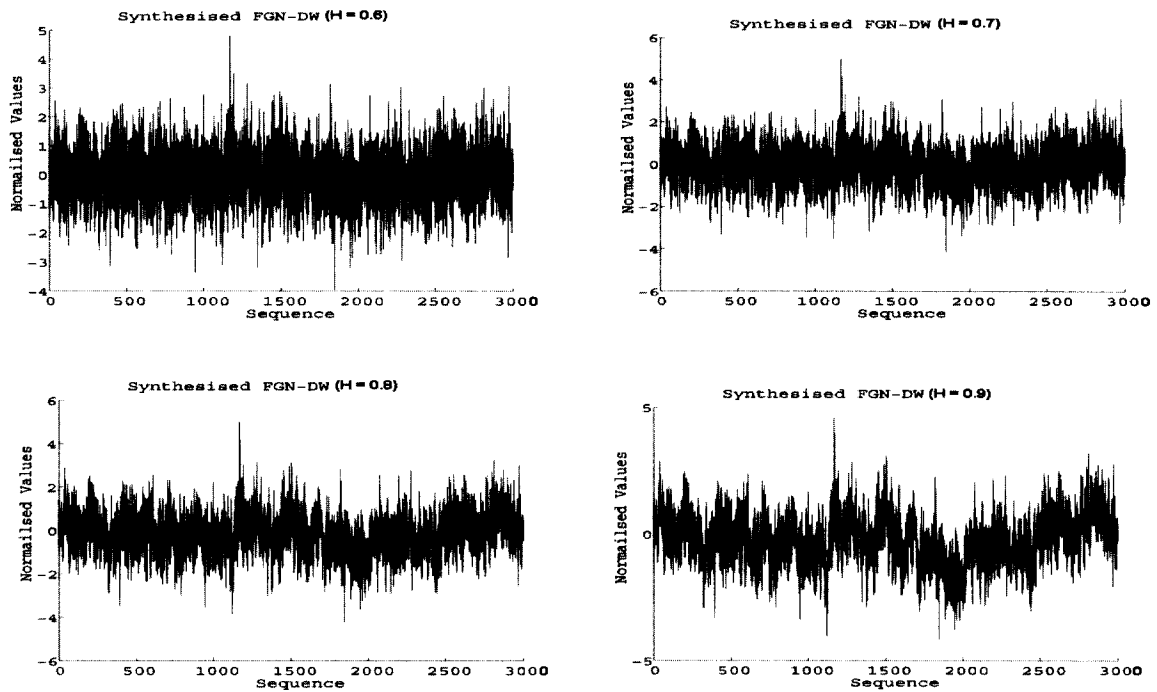
quences obtained by the FGN-DW method and the theoretical ACFs from Equation (5). As the H value increased, the ACF curves decayed hyperbolically and LRD was observed. For H = 0.6, 0.7, 0.8 and 0.9, relative inaccuracy ΔACF estimated from the ACF was -0.0208%, -0.0061%, -0.0076% and -0.0218%, respectively.

〈Table 4〉 Variances of estimated H obtained using Whittle's MLE for Daubechies wavelets and Haar wavelets for H = 0.6, 0.7, 0.8 and 0.9. Daub(♯) stands for the Daubechies wavelets with ♯ coefficients.

| Methods | Variances of Estimated H | | | |
	.6	.7	.8	.9
Haar	1.3111e-05	1.4501e-05	1.7235e-05	2.1660e-05
Daub(2)	1.1914e-05	1.2353e-05	1.3133e-05	1.4408e-05
Daub(4)	1.1653e-05	1.2084e-05	1.2638e-05	1.3323e-05
Daub(8)	1.1201e-05	1.1720e-05	1.2344e-05	1.3030e-05
Daub(16)	1.0946e-05	1.1519e-05	1.2156e-05	1.2812e-05

(Figure 2) Autocorrelation function plots for the FGN-DW method (H = 0.6, 0.7, 0.8 and 0.9)

(Figure 3) Sequence plots for the FGN-DW method (H = 0.6, 0.7, 0.8 and 0.9).

Sequence plots show stronger data correlation as the H value increased; see (Figure 3) for the FGN-DW method. In other words, generated sequences demonstrated evidence of LRD properties.

Our results show that the generator produces approximately self-similar FGN sequences, with the relative in-

accuracy, ΔH, increasing with the increase of H, but always staying below 8%. Apparently there is a problem with more detailed studies of such a generator, since different methods of analysis of the Hurst parameter can give different results for the bias of $\hat{H}$ in the same output sequences. More reliable methods for assessment of self-similarity in

pseudo-random sequences are needed.

6.4 Computational Complexity

The results of our experimental analysis of mean times needed by the generator for generating pseudo-random self-similar sequences of a given length are shown in <Table 5>. The main conclusion is that the FGN-DW method is fast. <Table 5> shows that 2 seconds were needed to generate a sequence of 32,768 (2^{15}) numbers, while generation of a sequence with 1,048,576 (2^{20}) numbers took 51 seconds. The theoretical algorithmic complexity of forming spectral density, and constructing normally distributed complex numbers, is $O(1)$, while the inverse DWT is $O(n)$ [16, 27]. Thus, the time complexity of FGN- DW is also $O(n)$.

In summary, our results show that a generator of pseudo-random self-similar sequences based on FGN and DW is sufficiently fast to make it applicable in practical computer simulation studies, when long self-similar sequences of numbers are needed.

〈Table 5〉 Complexity and mean running times of the FGN-DW generator. Running times were obtained by using the Matlab *clock* command on a Pentium II (233MHz, 64 MB) ; each mean is averaged over 30 iterations

Complexity	Sequence of				
	32,768 Numbers	131,072 Numbers	262,144 Numbers	524,288 Numbers	1,048,576 Numbers
	Mean running time (*minute : second*)				
$O(n)$	0 : 2	0 : 7	0 : 13	0 : 25	0 : 51

7. Conclusions

In this paper we have proposed a generator of (long) pseudo-random self-similar sequences, based on the FGN and DW transform. It appears that this generator produces approximately self-similar sequences, with the relative inaccuracy of the resulted H below 8%, if $0.6 \le H \le 0.9$. On the other hand, the analysis of mean times needed for generating sequences of a given length shows that this generator should be recommended for practical simulation studies of telecommunication networks, since it is very fast and accurate.

Acknowledgments

References

[1] W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *IEEE ACM Transactions on Networking*, Vol. 2, No.1, pp.1-15, 1994.

[2] N. Likhanov, B. Tsybakov and N.D. Georganas, "Analysis of an ATM Buffer with Self-Similar ('Fractal') Input Traffic," *Proceedings of IEEE INFOCOM '95*, Boston, Massachusetts, pp.985-992, 1995.

[3] V. Paxson and S. Floyd, "Wide-Area Traffic : the Failure of Poisson Modeling," *IEEE ACM Transactions on Networking*, Vol.3, No.3, pp.226-244, 1995.

[4] B. K. Ryu, "*Fractal Network Traffic : from Understanding to Implications*," PhD thesis, Graduate School of Arts and Sciences, Columbia University, 1996.

[5] M. W. Garrett and W. Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic," *Computer Communication Review, Proceedings of ACM SIGCOMM '94*, London, UK, Vol.24, No.4, pp.269-280, 1994.

[6] M. Krunz and A. Makowski, "A Source Model for VBR Video Traffic Based on $M/G/\infty$ Input Processes," *Proceedings of IEEE INFOCOM '98*, San Francisco, CA, USA, pp.1441-1448, 1998.

[7] O. Rose, "*Traffic Modeling of Variable Bit Rate MPEG Video and Its Impacts on ATM Networks*," PhD thesis, Bayerische Julius-Maximilians-Universitat Wurzburg, 1997.

[8] J. Beran, "Statistical Methods for Data with Long Range Dependence," *Statistical Science*, Vol.7, No.4, pp.404-427, 1992.

[9] A. L. Neidhardt and J. L. Wang, "The Concept of Relevant Time Scales and its Application to Queueing Analysis of Self-Similar Traffic (or Is Hurst Naughty or Nice?)," *Performance Evaluation Review, Proceedings of ACM SIGMETRICS '98*, Madison, Wisconsin, USA, pp.222-232, 1998.

[10] B. B. Mandelbrot, "A Fast Fractional Gaussian Noise Generator," Water Resources Research, Vol.7, pp.543-553, 1971.

[11] J. R. M. Hosking, "Modeling Persistence in Hydrological Time Series Using Fractional Differencing," *Water Resources Research*, Vol.20, No.12, pp.1898-1908, 1984.

[12] M. C. Cario and B. L. Nelson, "Numerical Methods for Fitting and Simulating Autoregressive-to-Anything Processes," *INFORMS Journal on Computing*, Vol.10, No.1, pp.72-81, 1998.

[13] T. Taralp, M. Devetsikiotis, I. Lambadaris and A. Bose, "Efficient Fractional Gaussian Noise Generation Using the Spatial Renewal Process," *Proceedings of IEEE International Conference on Communications (ICC '98)*, Atlanta,

GA, USA, pp.7-11, 1998.

[14] H.-D. J. Jeong, D. McNickle and K. Pawlikowski, "A Comparative Study of Three Self-Similar Teletraffic Generators," *Proceedings of 13th European Simulation Multiconference, ESM'99*, Vol.1, pp.356-362, Warsaw, Poland, 1999.

[15] W-C. Lau, A. Erramilli, J. L. Wang and W. Willinger, "Self-Similar Traffic Generation : the Random Midpoint Displacement Algorithm and its Properties," *Proceedings of IEEE International Conference on Communications (ICC'95)*, Seattle, WA, pp.466-472, 1995.

[16] V. Paxson, "Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic," *Computer Communication Review, ACM SIGCOMM*, Vol.27, No.5, pp.5-18, 1997.

[17] J. Beran, "*Statistics for Long-Memory Processes,*" Chapman and Hall, New York, 1994.

[18] K. Park and W. Willinger, "*Self-Similar Network Traffic and Performance Evaluation,*" Self-Similar Network Traffic : An Overview, John Wiley & Sons, Inc., K. Park and W. Willinger (eds), New York, pp.1-38, 2000.

[19] M. Taqqu, "Self-Similar Processes," *Encyclopedia of Statistical Sciences*, John Wiley and Sons, Inc., S. Kotz and N. Johnson (eds), New York, Vol.8, 1988.

[20] B. B. Mandelbrot and J. R. Wallis, "Computer Experiments with Fractional Gaussian Noises," *Water Resources Research*, Vol.5, No.1, pp.228-267, 1969.

[21] D. R. Cox, "Long-Range Dependence : a Review," *Statistics : An Appraisa, Iowa State Statistical Library*, The Iowa State University Press, H. A. David and H. T. David (eds.), pp.55-74, 1984.

[22] P. Flandrin, "Wavelet Analysis and Synthesis of Fractional Brownian Motion," *IEEE Transactions on Information Theory*, Vol.38, No.2, pp.910-917, 1992.

[23] L. M. Kaplan and C.-C. J. Kuo, "Fractal Estimation from Noisy Data via Discrete Fractional Gaussian Noise (DFGN) and the Haar Basis," *IEEE Transactions on Signal Processing*, Vol.41, No.12, pp.3554-3562, 1993.

[24] P. Abry and D. Veitch, "Wavelet Analysis of Long-Range-Dependent Traffic," *IEEE Transactions on Information Theory*, Vol.44, No.1, pp.2-15, 1998.

[25] S. Ma and C. Ji, "Modeling Video Traffic in the Wavelet Domain," *Proceedings of IEEE INFOCOM'98*, San Francisco, CA, USA, pp.201-208, 1998.

[26] I. Daubechies, "Ten Lectures on Wavelets," *CBMS-NSF Regional Conference Series in Applied Mathematics*, SIAM Press, Philadelphia, Pennsylvania, Vol.61, 1992.

[27] M. V. Wickerhauser, "*Adapted Wavelet Analysis from Theory to Software,*" A. K. Peters, Ltd., Wellesley, Massachusetts, 1994.

[28] H.-D. J. Jeong, D. McNickle and K. Pawlikowski, "Fast Self-Similar Teletraffic Generation Based on FGN and Inverse DWT," Tech. Rep. TR-COSC 03/99, Department of Computer Science, University of Canterbury, Christchurch, New Zealand, 1999.

[29] V. Ribeiro, R. Riedi, M. Crouse and R. Baraniuk, "Simulation of nonGaussian Long-Range-Dependent Traffic using Wavelets," *Performance Evaluation Review, Proceedings of ACM SIGMETRICS'99*, Vol.27, No.1, pp.1-12, 1999.

[30] 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.3, pp.992-1018, 1999.

[31] D. Veitch, J.-A. Bäckar, J. Wall, J. Yates and M. Roughan, "On-Line Generation of Fractal and Multifractal Traffic," *The PAM2000 Passive and Active Measurement Workshop*, Hamilton, New Zealand, 2000.

[32] M. Roughan and D. Veitch and P. Abry, "On-Line Estimation of the Parameters of Long-Range Dependence," *Proceedings of GLOBECOM'98*, Sydney, Australia, pp.3716-3721, 1998.

[33] S. J. Schiff, "Resolving Time-Series Structure with a Controlled Wavelet Transform," *Optical Engineering*, Vol.31, No.11, pp.2492-2495, 1992.

[34] S. Gonzalez Sanchez, N. Gonzalez Prelcic and S. Garcia Galan, "*Uvi_Wave. - Wavelets Toolbox for use with Matlab,*" Departamento de Tecnoloxias das Comunicacions. Universidade de Vigo, Vigo, ftp://ftp.tsc.uvigo.es/pub/Uvi_Wave.doc.ps.gz, 1996.

[35] D. Veitch and P. Abry, "A Wavelet Based Joint Estimator of the Parameters of Long-Range Dependence," *IEEE Transactions on Information Theory*, Special Issue on Multiscale Statistical Signal Analysis and its Applications, Vol.45, No.3, pp.878-897, 1999.

[36] M. Roughan and D. Veitch, "Measuring Long-Range Dependence Under Changing Traffic Conditions," *Proceedings of IEEE INFOCOM'99*, New York, NY, USA, pp.1513-1521, 1999.

[37] H.-D. Jeong, D. McNickle and K. Pawlikowski, "An Empirical Comparison of Hurst Parameter Estimators," *IEEE Transactions on Information Theory*, Submitted, June, 2003.

Appendix A : A Program for Generating Self-Similar Sequences

Here is a set of Matlab functions for implementing the method, based on FGN and Daubechies wavelets, described in this paper.

```
% This function returns a self-similar sequence with n
numbers and the Hurst parameter H.
function SS = FGNDW(n, H, Scale, VanishingMoment)

% Create n frequencies, then calculate fast, approximately
power spectrum.
lambda = ((1 : n)×pi)/n ;
f = FGNDWSpectrum(lambda, H) ;

% Adjust for estimating power spectrum via periodogram.
r = random('Exponential', 1, 1, n) ;
f_r = f.×r ;

% Construct corresponding complex numbers with random
phase.
re = sqrt(f_r) ;
im = random('Uniform', 0, 2*pi, 1, n) ;

% Calculate complex spectral density and real part/imagi-
nary part form.
real_part = re.×cos(im) ;
imag_part = re.×sin(im) ;
z = real_part + imag_part×i ;

% Calculate filter values using I. Daubechies' algorithm.
[h, g, rh, rg] = daub(VanishingMoment) ;

% Calculate a sequence in time domain using inverse dis-
crete wavelet transform (IDWT).
SS = real(iwt(z, rh, rg, Scale)) ;

%----------------------------------------------
% Returns an approximation of the power spectrum for
Fractional Gaussian Noise at the given
```

```
% frequencies lambda and the given Hurst parameter H.
% ----------------------------------------------
function FGNS = FGNDWSpectrum(lambda, H)

cf = (1/(2×pi))×var(lambda)×sin(pi×H).×gamma(2×H+1) ;
FGNS = cf×(abs(lambda)).^(1-2×H) ;
```

정 해 덕

e-mail : joshua@bible.ac.kr
1997년 Third Wave Media Ltd., NZ, 미디어 통계분석가
1999년~2003년 Univ. of Canterbury, NZ, 연구원
2002년 Dept. of Computer Science, University of Canterbury,
    NZ, 컴퓨터과학 박사
2003년~2004년 Quality Education Support Service in NZ Ltd.,
    NZ, 연구부장
2004년~2004년 Sarona Community Trust, NZ, 개별지도교수
2004년~현재 한국성서대학교 정보과학부 교수
관심분야 : Teletraffic Modeling & Analysis in Telecommuni-
    cation Networks and Stochastic Simulation

이 종 숙

e-mail : jsruthlee@kisti.re.kr
1992년~1993년 한국전자통신연구원
    연구원
1999년~2002년 Univ. of Canterbury, NZ,
    연구원
2001년 Dept. of Computer Science,
    University of Canterbury, NZ,
    컴퓨터과학 박사
2002년~현재 한국과학기술정보연구원 그리드연구실, 선임연구원
관심분야 : 그리드 컴퓨팅, 그리드 미들웨어, 병렬/분산 컴퓨팅,
    분산/병렬 시뮬레이션