

# Wavelet-Based Resource Allocation in ATM Networks

Patrick Droz, dro@zurich.ibm.com  
IBM Research Division, Zürich Research Laboratory,  
8803 Rüschlikon, Switzerland  
Tel. +41-1-724-85-25, Fax. +41-1-710-36-08

**ABSTRACT** In recent years, numerous publications have identified a high number of traffic sources to be of self-similar nature. Such traffic can have a negative impact on traditional queuing systems. This paper presents a call admission control algorithm that uses periodic wavelet analysis applied to traffic measurements to estimate the amount of resources necessary to support a certain set of connections under quality-of-service constraints. This algorithm even exploits correlated traffic. Extensive simulation results illustrate the performance of the algorithm over a large parameter spectrum (Hurst parameter, load, and intensity), and the results consistently show that the higher the Hurst parameter, the higher the statistical gain.

## 1 Introduction

In an *asynchronous transfer mode* (ATM) network [Uni94, Uni93, Atm93] a user can specify additional traffic parameters, e.g. peak cell rate and sustainable cell rate, that describe the behavior of the traffic generation pattern in more detail. Based on these values the *call admission control* (CAC) function accepts or rejects the new connection. In the case of acceptance the *resource manager* (RM) allocates a certain amount of resources. The key question is exactly how much “a certain amount” has to be in order to fulfill the given *quality-of-service* (QoS) requirements. As the peak cell rate is mandatory for every connection it is of course possible to make a peak rate allocation. On the other hand peak rate allocation tends to waste resources. Therefore, a tradeoff between customer satisfaction and resource utilization has to be made. This task is extremely difficult in the presence of bursty traffic. In recent years, many publications [Lel94, Lel93] have revealed and analyzed the self-similar (fractal) nature of many traffic types. Self-similar traffic is bursty and the aggregation of such traffic streams exhibits a slower smoothing effect. Queuing systems designed by means of traditional traffic models may therefore be under-dimensioned and may lead to higher cell loss rates than originally predicted.

A highly adaptive, measurement-based CAC algorithm that can cope with and even exploit self-similar traffic is introduced and assessed. The algorithm attempts to derive an effective capacity from periodic traffic measurements. This estimation method employs periodic wavelet analysis applied to overlapping windows of traffic measurements. Extensive simulation results will be presented that demonstrate the performance of the algorithm, i.e. the algorithm is evaluated

in terms of the Hurst parameter with which one can alter the grade of self-similarity.

This paper is structured as follows. In Section 2 the traffic simulator used to assess the performance of the CAC will be briefly presented. Section 3 will introduce wavelets. Section 4 describes and illustrates the CAC, including the resource estimation model. Section 5 presents extensive simulation results. The final Section 6 summarizes the paper and provides an outlook.

## 2 Traffic Generator

Numerous methods are known in the literature to produce self-similar traffic. Most of these models, however, are used for connectionless traffic and therefore do not produce corresponding traffic parameters such as the peak and sustainable cell rate used in ATM. For this reason the *random midpoint displacement* (RMD) algorithm [Lau95, Pei92, Fou82] was extended into a two-level process that generates on the first level the traffic parameters and on the second level the traffic

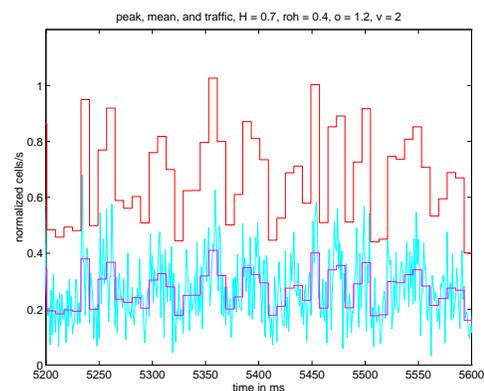


Figure 1: Traffic generator output

itself. A detailed description of the algorithm can be found in [Dro96b, Dro96c]. Because the traffic generator is used in this paper as a tool, only the in- and output parameters are described. The set of input parameters is  $N$ , number of samples  $2^N$ ,  $H$ , Hurst parameter ( $0 < H < 1$ ),  $\rho$ , sustainable to peak cell rate ( $0 < \rho \leq 1.0$ ),  $s$ , starting seed for the random number generation,  $o$ , maximal overbooking (how far the aggregated peak cell rate can exceed the link capacity),  $v$ , noise intensity (scaling of the actual traffic around the aggregated sustainable cell rate).

A snapshot of generated traffic parameters and traffic is depicted in Fig. 1. The output is regarded as the aggregation of a certain set of connections, where the changes of reservations are discretized to equidistant points in time. For a 155 Mbit/s link, we generated a traffic sample every millisecond. Changes in the reservation state occurred every 8 ms.

### 3 Introduction to Wavelets

This section introduces briefly wavelets [Gra95]. A profound background of the theory can be found in [Kai94].

Wavelet analysis can be compared to Fourier analysis. It transforms a signal from the time domain into the frequency domain. In the case of Fourier analysis, data is either in the time domain or in the frequency domain but there is no direct relationship between the two. This means that in the frequency domain one can determine which frequencies appeared and how strongly they are present but not where (in the time domain) they appeared. To cope with this problem a *windowed fourier transformation* (WFT) can be used. It splits the input signal into different windows, which are then analyzed individually. The difficulty of this approach lies in the way the window borders have to be handled. The wavelet analysis solves this problem by dealing with the input signal at different scales or resolutions. On a large scale the focus is on rough features, i.e. trends, whereas on a small scale minor features, i.e. noise, are of interest.

The general idea is to adopt a wavelet prototype function, called an *analyzing wavelet* or a *mother wavelet*. These functions are bases in the function space. Compared to a Fourier transform where only one basis consisting of sines and cosines of different frequencies exists, one has the choice among numerous bases. This freedom has its price of course, as it is often quite difficult to find the optimal basis for a particular input data set. Temporal analysis is carried out with a contracted high-frequency version of the mother wavelet. The frequency analysis is done with a dilated low-frequency version of the prototype wavelet.

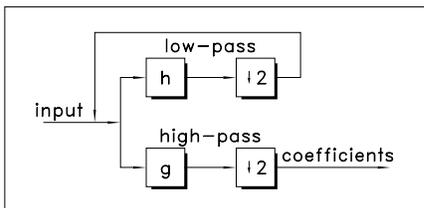


Figure 2: Fast wavelet transformation algorithm

Similar to the *fast Fourier transformation* (FFT) there exists a *fast wavelet transformation* (FWT). The FFT has a complexity of  $O(n \log n)$ , whereas the FWT can be calculated in  $O(n)$ . For both transformations the input length is expected to be a power of 2, i.e.  $2^n$ . Figure 2 is a block diagram of the FWT algorithm. The box marked  $\downarrow 2$  stands for a down-sampling of 2. This is achieved by dropping every other sample of the input data. The high-pass filtered output

on each scale is kept as the coefficients of that scale. The low-pass filtered and down-sampled output is fed back as input to the transformation process. This process is sometimes called a *pyramid algorithm* because of its hierarchical processing. The loop is performed until no samples are left. On each scale the number of coefficients is reduced by a factor of 2. The total number of coefficients over all scales equals the number of input values. Low-pass filtering smooths the signal, whereas high-pass filtering reveals details. The synthesis is similar to the analysis but, instead of a down-sampling afterwards, first an up-sampling is carried out by inserting a zero every other sample.

The simplest basis known is the Haar wavelet, called after its inventor. The scale function is a moving average operator and the wavelet function is a moving differentiator.

The most striking difference between FFT and FWT is that the wavelet functions are localized in space and frequency. Sines and cosines do not have both properties. Because they are periodic nonvanishing functions, they are localized only in frequency. The twofold localization frequently generates a sparse coefficient vector.

To illustrate the twofold localization for the Haar wavelet, Fig. 3 gives an example with 8 samples. On the first scale the coefficients  $c_1$  to  $c_4$  are generated. Then on the second scale,  $c_5$  and  $c_6$  are generated. The underbracing indicates the localization of the calculated coefficient. This region is given

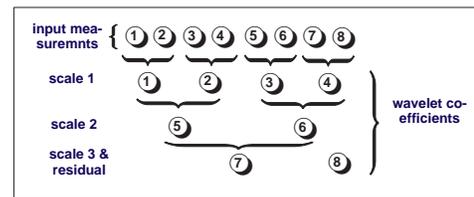


Figure 3: Wavelet hierarchy

by the down-sampling operation that compresses the data in each iteration by a factor of 2. Compared to the FFT the coefficients of the FWT belong to a certain part of the input signal. In most implementations the coefficients are put in one vector one scale after the other, the last coefficient being the final residual of the low-pass filtering.

### 4 Wavelet-based CAC

Figure 4 depicts the building blocks of the CAC. For simplicity the description it is assumed that only one instance of the method is present. In general, multiple instances that control groups or individual connections can be present. The method is a hop-by-hop architecture, which has to be applied to every leg of the path. Some connections can be excluded from the method.

Cells arriving from the switch are put into the link buffer. The sampler generates periodic traffic measurements. The measurements consist of cell counts divided by  $\Delta t$ . The sampling frequency depends on the link speed and the buffer size. The sampler feeds its output into the buffer of the *digital signal processor* (DSP). In periodic intervals the DSP analyzes

the data and calculates the effective capacity, which it sends to the resource manager. The detailed algorithm can be found in [Dro96a, Dro96c]. The queue manager checks the queue length. If the length exceeds a certain threshold or

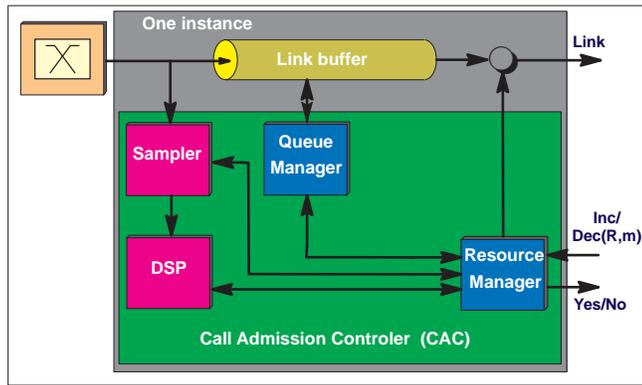


Figure 4: General architecture

if cell losses occur, this is signaled to the resource manager. The queue manager also signals to the resource manager if a certain period of time passes without cell losses. Call arrivals and departures as well as increments or decrements are presented to the resource manager by the signaling protocol, e.g. Q.2931. The resource manager either accepts or rejects arrivals and increases. Departures and decreases are always granted. It also adjusts the out rate of the buffer.

The signal processor calculates the effective capacity. First, a wavelet transformation is applied to the series of samples. Then the coefficients are manipulated and back-transformed. From the reverse transformed signal the effective capacity is derived and used as a starting point for the reservation process. For new connections peak rate allocation is performed first. After the observation phase the connections become part of a global effective capacity. In this context “global” refers to the grouping used, e.g. a virtual path or a specific traffic class.

A basis-independent estimation of the effective capacity is illustrated in Fig. 5. First the wavelet coefficients are calculated. Then the coefficients are manipulated in the sense that we count the highest two scales to the noise. Afterwards, the noise is obtained by the reverse transformation. At this point in time the following equation holds: Original traffic measurements – Noise = Separated signal. Then the cumulative sum curve of the noise is calculated, which is regarded as the theoretical buffer filling during the course of this time window by assuming a buffer output rate according to the separated signal. In other words, buffers are never occupied by the separated signal; only noise uses buffers. The threshold for the number of scales that are counted to the noise is set based on the cumulative sum such that the maximum value would occupy only a small amount of buffers, e.g. 10%. As soon as the threshold is determined, the separated signal is calculated and the effective capacity is taken as the maximum in this window. The separation

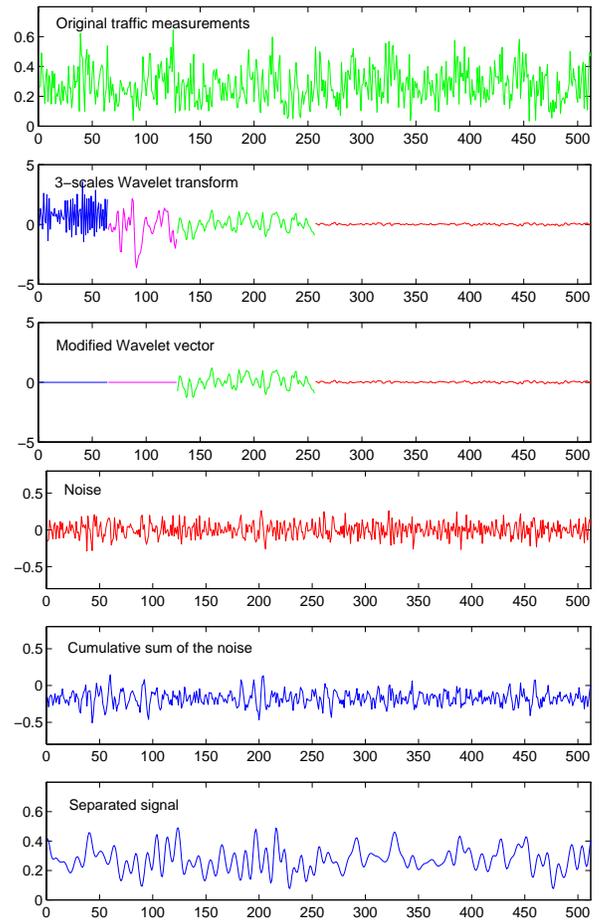


Figure 5: Estimation of effective capacity

threshold is determined dynamically for every overlapping window.

## 5 Extensive Simulation Results

In this section the simulation results and their interpretation will be presented. Because the exact values of the Hurst parameter, the utilization  $\rho$  ( $m/R$ ), and the noise reduction are still unknown, the simulations were carried out over a wide parameter spectrum in order to assess the adaptability of the CAC algorithm.

The Hurst parameter was varied between 0.4 (slightly negative correlation) and 0.9 (strong positive correlation). Because in communication systems only positive correlations have been observed, we focus on that region. For noise intensity an interval of between 1 (the aggregated traffic is regarded as a single on/off source) and 9 (very smooth traffic) was covered. The utilization  $\rho$  was varied between 0.1 and 0.9.

Figures 6-8 show the statistical gain ( $1 - \text{ratio of effective capacity to peak rate}$ ) including the confidence intervals for 95% confidence with fixed Hurst parameter. Every crossing point of the mesh was calculated with at least  $10^{10}$  simulated cells. The behavior of the CAC has the following important property. Towards the borders the confidence intervals become very tight, which indicates the convergence of

the CAC. For  $\rho = 0.1$  this is because the algorithm reaches the maximum overbooking allowed. For  $\rho = 0.9$  the algo-

rate allocation this is an enormous gain. Especially on expensive leased lines this CAC can increase the utilization

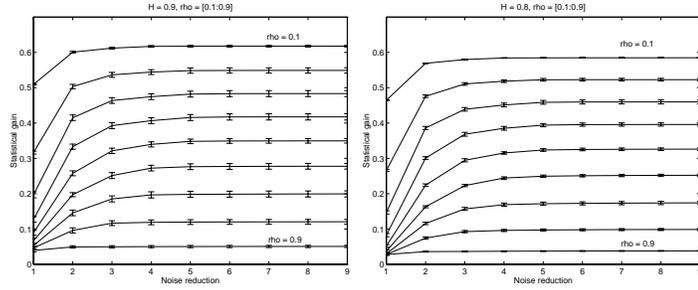


Figure 6: Stat. gain,  $H = 0.8$  and  $0.9$ , (95% confidence)

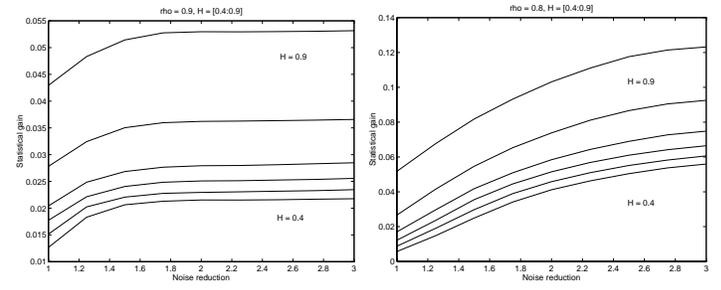


Figure 9: Statistical gain for  $\rho = 0.9$  and  $\rho = 0.8$

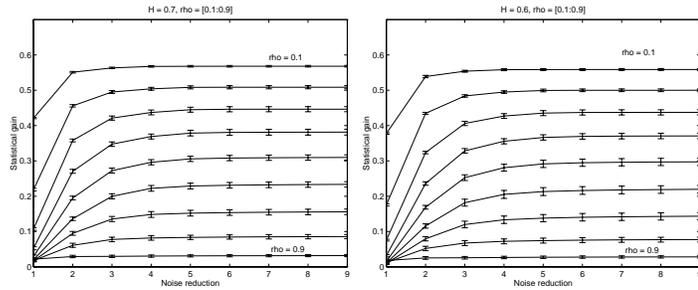


Figure 7: Stat. gain,  $H = 0.7$  and  $0.6$ , (95% confidence)

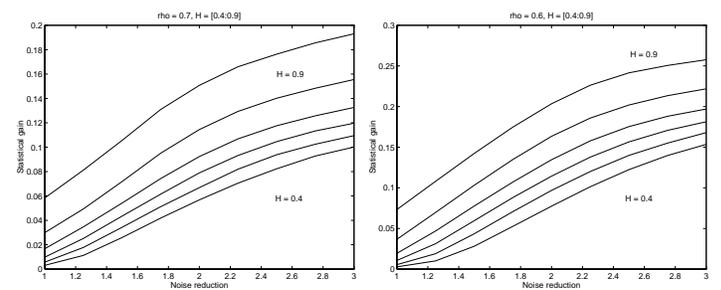


Figure 10: Statistical gain for  $\rho = 0.7$  and  $\rho = 0.6$

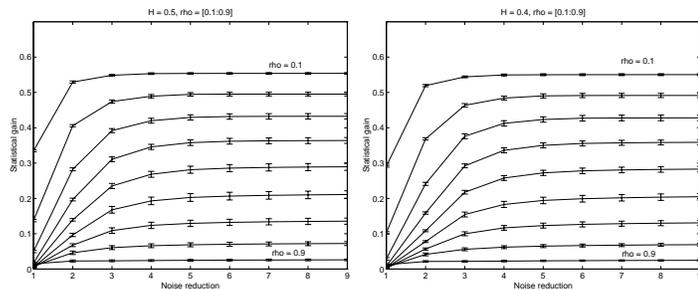


Figure 8: Stat. gain,  $H = 0.5$  and  $0.4$ , (95% confidence)

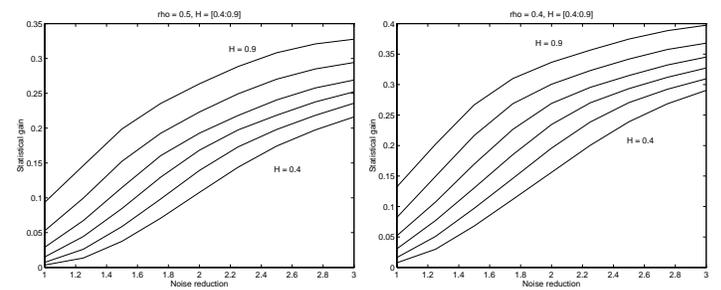


Figure 11: Statistical gain for  $\rho = 0.5$  and  $\rho = 0.4$

gorithm tends to peak rate allocation because of heavy traffic.

The calculation time per run per picture was about one week on an IBM RS6000 Model 580. Five runs were made per picture. With the chosen buffer utilization of 0.1 and the maximum link utilization of 0.8, all runs were subject to zero cell loss. In order to run through the entire parameter spectrum a total of about 30 weeks of full-time simulation was necessary. In Figs. 9-12 zooms are depicted with a restricted interval for  $\rho$  but with the entire Hurst parameter spectrum. This interval contains the operational part of the algorithm with the highest dynamics. The statistical gain ranges from 0% to slightly more than 60%. These two extremes are unlikely to be representative of average networking traffic. The probability of  $\rho$  being between 0.3 and 0.7 is much higher. The statistical gain for this operational region is between 10% and 48% the average being 29%. Compared to peak

significantly, thus reducing link costs drastically.

Figs. 9-12 illustrate very consistently that the higher the Hurst parameter, the higher the statistical gain, independent of noise intensity or utilization. These simulation results reveal that the wavelet-based CAC exploits correlation structures. In addition, they confirm the adaptivity of the algorithm.

The starting point of each curve already includes a statistical gain. The reason for this is that the maximum link utilization is set to 0.8, thus adding a safety margin to the allocation process.

The reasons for exploiting the correlations are numerous. One is that, in the case of positive correlation, a more accurate prediction can be made because the probability that an increasing interval will be followed by another increasing interval is higher than the probability that an increasing

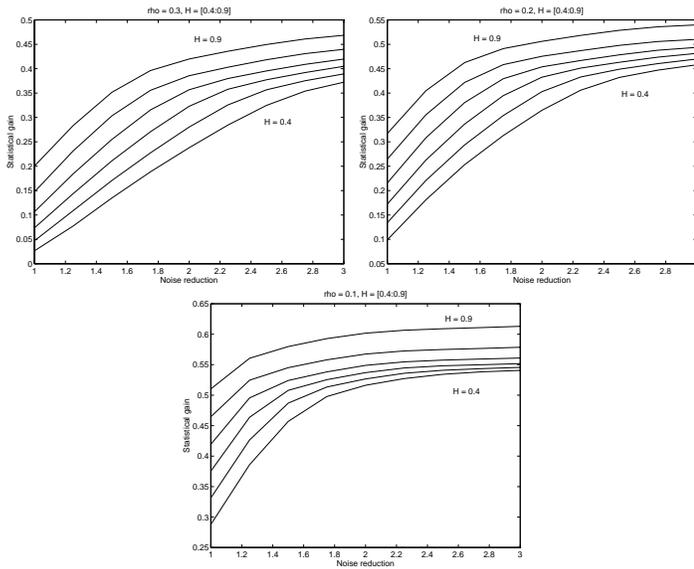


Figure 12: Statistical gain for  $\rho = 0.3$ ,  $\rho = 0.2$ , and  $\rho = 0.1$

interval is followed by a decreasing interval and vice versa. This causes a kind of trend line to which the algorithm can adapt itself. The trend line is captured by the wavelet transformation from which the effective capacity is derived and estimated for the next interval. Wavelet analysis is a highly appropriate tool because the input signal is analyzed on different time scales, thus existing correlation structures can be captured. Theoretically it would again be possible to have better prediction with  $H$  values below 0.5 but the changes are too short-term and, therefore, cannot be accommodated.

Simulations revealed a striking difference between the maximum and average buffer occupancy. The average buffer utilization was extremely low, whereas the maximal filling reached values of about one third of the available buffer slots. This is an intrinsic property of self-similar traffic because bursts occur over a wide range of time scales.

## 6 Summary & Outlook

A CAC has been introduced that uses wavelet-based traffic estimation to derive an effective capacity to support a certain set of connections. The wavelet analyses are periodically applied to overlapping windows of traffic measurements. The CAC was evaluated with self-similar traffic over a large parameter spectrum. It was shown that the CAC achieves a higher statistical gain, the higher the positive correlation. This result contradicts those of other publications, which predict that self-similar traffic will have a negative impact on queuing systems.

Numerous research activities are needed for the future. Real-world measurements have to be analyzed to substantiate the assumption of the self-similar nature of ATM traffic. Publications on this topic are forthcoming. Highly appropriate wavelet basis are to be determined and more sophisticated coefficient manipulations and estimation methods remain to be done.

## References

- [Atm93] The ATM Forum. *ATM user-network interface specification, version 3.0*. PTR Prentice Hall, 1993. ISBN 0-13-225863-3.
- [Dro96a] Patrick Droz. *Traffic Estimation and Resource Allocation based on Periodical Wavelet Analysis*. 4th Australian International Symposium on DSP for Communication Systems, September 1996.
- [Dro96b] Patrick Droz and Jean-Yves Le Boudec. *A High-Speed Self-Similar ATM VBR Traffic Generator*. GLOBECOM 96, London, November 1996.
- [Dro96c] Patrick W. Droz-Georget. *Traffic Estimation and Resource Allocation in ATM Networks*. PhD Thesis No. 11462, Swiss Federal Institute of Technology, 1996.
- [Fou82] A. Fournier, D. Fussell, and L. Carpenter. *Computer Rendering of Stochastic Models*. ACM Comm 25, pages 371–384, 1982.
- [Gra95] Amara Graps. *An Introduction to Wavelets*. IEEE Computational Science and Engineering, 2(2), 1995. <http://www.best.com/~agraps/papers/papers.html>.
- [Kai94] Gerald Kaiser. *A Friendly Guide to Wavelets*. Birkhäuser, 1994. ISBN 0-8176-3711-7, 3-7643-3711-7.
- [Lau95] Wing-Cheong Lau, Ashok Erramilli, Jonathan L. Wang, and Walter Willinger. *Self-Similar Traffic Generation: The Random Midpoint Displacement Algorithm and Its Properties*. IEEE ICC, June 1995.
- [Lel93] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. *On the Self-Similar Nature of Ethernet Traffic*. ACM SIGCOM, September 1993.
- [Lel94] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. *On the Self-Similar Nature of Ethernet Traffic (Extended Version)*. IEEE/ACM Transactions On Networking, 2(1), February 1994.
- [Pei92] Heinz-Otto Peitgen, Hartmut Jürgens, and Dietmar Saupe. *Chaos and Fractals, New Frontiers of Science*. Springer Verlag, 1992. ISBN 0-387-97903-4, ISBN 3-540-97903-4.
- [Uni93] Richard Breault, Jim Grace, John Jaeger, and Lou Wojnaroski. *Atm user-network interface specification version uni 3.0*. Technical report, ATM Forum, August 1993.
- [Uni94] G. Ratta. *ATM User-Network Interface Specification Version UNI 3.1*. ATM Forum, July 1994.