

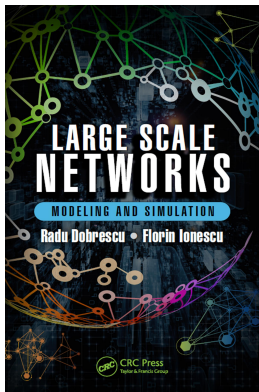
This article was downloaded by: 10.2.98.160

On: 26 Oct 2020

Access details: *subscription number*

Publisher: *CRC Press*

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: 5 Howick Place, London SW1P 1WG, UK



Large Scale Networks: Modeling and Simulation

Radu Dobrescu, Florin Ionescu

Introduction

Publication details

<https://test.routledgehandbooks.com/doi/10.1201/9781315368368-4>

Radu Dobrescu, Florin Ionescu

Published online on: 10 Oct 2016

How to cite :- Radu Dobrescu, Florin Ionescu. 10 Oct 2016, *Introduction from: Large Scale Networks: Modeling and Simulation* CRC Press

Accessed on: 26 Oct 2020

<https://test.routledgehandbooks.com/doi/10.1201/9781315368368-4>

PLEASE SCROLL DOWN FOR DOCUMENT

Full terms and conditions of use: <https://test.routledgehandbooks.com/legal-notices/terms>

This Document PDF may be used for research, teaching and private study purposes. Any substantial or systematic reproductions, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The publisher shall not be liable for an loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

chapter three

Self-similarity in traffic

3.1 Self-similar traffic and network performance

One of the most significant findings of traffic measurement studies over time has been the observed self-similarity in packet network traffic. Subsequent research has focused on the origins of such self-similarity, and the network engineering significance of this phenomenon, namely, the manner in which network dynamics (specifically, the dynamics of transmission control protocol [TCP], the predominant transport protocol used in today's Internet, which in turn exhibits mainly multimedia traffic) can affect the observed self-similarity.

Models of multimedia traffic offered to the network or to a component of the network are critical in providing high quality of service (QoS). Traffic models are used as the input to analytical or simulation studies of resource allocation strategies. We may view traffic at the application or packet level, where an application-level view may simply describe the profiled traffic as "a videoconference between three parties," while the packet-level view is based on a stochastic model that mimics the arrival process of packets associated with this application reasonably well. Clearly, in order to quantify traffic, packet-level representation of applications will be used. An important feature of multimedia traffic at the packet level having a significant impact on performance is traffic correlation. The complexity of traffic in a multimedia network is a natural consequence of integrating, over a single communication channel, a diverse range of traffic sources such as video, voice, and data that significantly differ in their traffic patterns as well as their performance requirements. Specifically, "bursty" traffic patterns generated by data sources and variable bit rate (VBR) real-time applications such as compressed video and audio tend to exhibit certain degrees of correlation between arrivals, and show LRD in time (Sahinoglu and Tekinay 1999).

The questions that arise here are how prevalent such traffic patterns are and under what conditions is performance analysis critically dependent upon taking self-similarity into account. There are different studies pointing out either the importance of self-similarity to network performance or the irrelevance of the need for capturing self-similarity in traffic modeling. To clarify this dilemma, a thorough understanding of QoS and resource allocation in a network environment is necessary. An optimal resource allocation would mean determining optimal buffer sizes,

assignment of bandwidth, and other resources in order to get the desired QoS expressed in terms of parameters such as queuing delay, retransmission time, packet loss probability, and bit error rate.

3.1.1 *Quality of service and resource allocation*

The International Organization for Standardization (ISO) defines QoS as a measure for denoting how good the offered networking services are. Generally, QoS parameters are performance measures such as bit error rate, frame error rate, cell loss probability, delay, and delay variation or guarantee, which is the maximum difference between end-to-end delays experienced by any two packets, are used as the determining QoS parameters. The user and application requirements for the MCS are mapped into a communication system that tries to satisfy the requirements of the services, which are parameterized.

Parameterization of the services is defined in ISO standards through the notion of QoS. The set of chosen parameters for a particular service determines what will be measured as the QoS. Network QoS parameters describe requirements for network services. They may be specified in terms of (a) network load, characterized by average/minimal interarrival time on the network connection, packet cell size and service time in the node for the connection's packet/cell and (b) network performance, describing the requirements which the network services have to guarantee. The performance might be expressed through a source-to-destination delay bound for the connection's packet loss rate.

Services for multimedia networked applications need resources to perform their functions. Of special interest in the study of network characteristics that are shared among the application, system, and network. There are several constraints that must be satisfied during multimedia transmission: time constraints, which include delays, computing time, and signaling delay; space constraints such as system buffers; and frequency constraints like network bandwidth and system bandwidth for data transmission (Nahrstedt and Steinmetz 1994). The best utilization of resources in a network environment is only possible by first characterizing the traffic and then determining the parameters such as buffer size and bandwidth to maximize performance. The main objective in telecommunications network engineering is to have as many happy users as possible. In other words, the network engineer has to resolve the tradeoff between capacity and QoS requirements. Accurate modeling of the offered traffic load is the first step in optimizing resource allocation algorithms such that provision of services complies with the QoS constraints while maintaining maximum capacity. In recent years, as broadband multimedia services became popular, they have necessitated the development of new traffic models with self-similar characteristics.

3.1.2 Concept of self-similarity

A self-similar phenomenon displays structural similarities across a wide range of timescales. Traffic that is bursty on many or all timescales can be described statistically using the notion of self-similarity. Self-similarity is the property associated with “fractals,” which are objects whose appearances remain unchanged regardless of the scale at which they are viewed. In the case of stochastic objects like time series, self-similarity is used in the distributed sense: when viewed at varying timescales, the object’s relational structure remains unchanged. As a result, such a time series exhibits bursts at a wide range of timescales.

In 1993, paper titled “On the Self-Similar Nature of Ethernet Traffic” (Leland et al. 1993) was published, considered a landmark in the field of network performance modeling. Ethernet is a broadcast multiaccess system for local area networking with distributed control. The authors reported the results of a massive study of Ethernet traffic and demonstrated that it had a self-similar (i.e., fractal) characteristic. This meant the Ethernet traffic had similar statistical properties at a range of timescales: milliseconds, seconds, minutes, hours, and even days and weeks. Another consequence is that the merging of traffic streams, as in a statistical multiplexer or an asynchronous transfer mode (ATM) switch, does not result in smoothing of the traffic. Again, bursty data streams that are multiplexed tend to produce a bursty aggregate stream. This first paper sparked a surge of research around the globe. The results show the self-similarity in ATM traffic, compressed digital video streams, and Web traffic between browsers and servers. Although a number of researchers had observed over the years that network traffic did not always obey Poisson assumptions applied in queuing analysis, for the first time, this paper provided an explanation and a systematic approach to modeling realistic data traffic patterns. Following the characterization of the fractal nature of data traffic, network theorists were split into two camps, with one group advocating the rewriting of and the other disagreed to this proposal. Traditionally, networks have been described by generalized Markovian processes, which are statistical models relying on postulates framed by the Russian mathematician A. A. Markov. Markovian models of networks have limited memory of the past. They reflect SRD. In a Markovian model, smoothing of bursty data is possible. Averaging of bursty traffic over a long period of time gives rise to a smooth data stream. A network based on fractal nature will have very different parameters and congestion control techniques.

X is defined to be a wide-sense stationary random process with the mean m , variance represented by s , and autocorrelation function denoted by r . For each m , $X(m)$ defines a wide-sense stationary random process.

The process X is said to be second-order self-similar with self-similarity parameter H if the aggregated processes have the same

autocorrelation structure as X . In other words, X is exactly second-order self-similar if the aggregated processes are indistinguishable from X with respect to their first- and second-order properties. The most striking feature of self-similarity is that the correlation structures of the aggregated process do not degenerate as $m \rightarrow \infty$. This is in contrast to the traditional models, all of which have the property that the correlation structure of their aggregated processes degenerates as $m \rightarrow \infty$. The Hurst parameter H is a measure of the level of self-similarity in a time series. H takes values from 0.5 to 1. In order to determine if a given series exhibits self-similarity, a method is needed to estimate H for a given series. Currently, there are three approaches to measuring H : analysis of the variances of the aggregated processes $X(m)$; analysis of the rescaled range (R/S) statistic for different block sizes; a Whittle estimator.

Since self-similarity is believed to have a significant impact on network performance, understanding the causes of self-similarity in traffic is important. In a realistic client/server network environment, the degree to which file sizes are heavy-tailed can directly determine the degree of traffic self-similarity at the link level. This causal relation is proven to be robust with respect to changes in network resources (bottleneck bandwidth and buffer capacity), network topology, the influence of cross-traffic, and the distribution of interarrival times. Specifically, for measuring self-similarity via the Hurst parameter H and the file size distribution by its power-law exponent α , it has been shown that there is a linear relationship between H and α over a wide range of network conditions.

3.1.3 *Effects of self-similarity on network performance*

Well-defined metrics of delay, packet loss, flow capacity, and availability are fundamental to measurement and comparison of path and network performance. In general, users are most interested in metrics that provide an indication of the likelihood that their packets will reach the destination in a timely manner. Therefore, the estimates of past and expected performance for traffic across specific Internet paths, and not simply measures of current performance, are important. Users are also increasingly concerned about path availability information, particularly as it affects the quality of multimedia applications requiring higher bandwidth and lower latency, such as Internet phone and videoconferencing. Availability of such data could help in scheduling online events such as Internet-based distance education seminars, and also influence user willingness to purchase higher service quality and associated service guarantees.

Given the ubiquity of scale-invariant burstiness observed across diverse networking contexts, finding effective traffic control algorithms

capable of detecting and managing self-similar traffic has become an important problem. The control of self-similar traffic involves modulating the traffic flow in such a way that the resulting performance is optimized. Scale-invariant burstiness (i.e., self-similarity) introduces new complexities into optimization of network performance and makes the task of providing QoS together with achieving high utilization difficult. Many analytical studies have shown that self-similar network traffic can have a detrimental impact on network performance, including amplified queuing delay and packet loss rate. On the other hand, LRD is unimportant for buffer occupancy when the Hurst parameter is not very large ($H < 0.7$). One practical effect of self-similarity is that the buffers needed at switches and multiplexers must be bigger than those predicted by traditional queuing analysis and simulations. The delay-bandwidth product problem arising out of high-bandwidth networks and QoS issues stemming from the support of real-time multimedia communication have added further complexities to the problem of optimizing performance.

How much self-similarity affects network performance is modulated by the protocols acting at the transport-network layer. An exponential tradeoff relationship was observed between queuing delay and packet loss rate. It is certain that a linear increase in buffer sizes will produce a nearly exponential decrease in packet loss, and that an increase in buffer size will result in a proportional increase in the effective use of transmission capacity. With self-similar traffic, these assumptions do not hold. The decrease in packet loss with buffer size is far lower than expected and the buffer requirements begin to explode at the lower levels of utilization for higher degrees of LRD (higher values of H). Moreover, scale-invariant burstiness implies the existence of concentrated periods of high activity at a wide range of timescales, which adversely affects congestion control and is an important correlation structure, which may be exploitable for congestion control purposes. Network performance as captured by throughput, packet loss rate, and packet retransmission rate degrades gradually with increasing heavy-tailedness. The degree to which heavy-tailedness affects self-similarity is determined by how well congestion control is able to shape its source traffic into an on-average constant output stream while conserving flow.

Packet loss and retransmission rate decline smoothly as self-similarity increases in the condition of a reliable flow-controlled packet transport. The only performance indicator exhibiting a more sensitive dependence on self-similarity is mean queue length, and this concurs with the observation that queue length distribution under self-similar traffic decays more slowly than with Poisson sources. Increasing network resources such as link bandwidth and buffer space results in a linear improvement in performance. However, large buffer sizes are accompanied by long queuing delays. In the context of facilitating multimedia traffic such as video and

voice in a best-effort manner while satisfying their diverse QoS requirements, low packet loss, on average, can be achieved only with a significant increase in queuing delay and vice versa. Increasing link bandwidth, given a large buffer capacity, has the effect of decreasing queuing delay much more drastically under highly self-similar traffic conditions than when traffic is less self-similar.

3.2 Mathematics of self-similar processes

3.2.1 Stationary random processes

To describe the character of the observed scaling properties of traffic data more precisely, we introduce a set of definitions on first- and second-order stationary time series.

Let consider a time series $X = X_n, n \in Z^+$, which represents the number of packets X_n that arrive in the n th interval of time T_i . Therefore,

$$X_n = N[nT_i] - N[(n-1)T_i] \quad (3.1)$$

The number of packets arrival till the t moment represents a counting process given by

$$N(t) = \int_0^t dN(t) \quad (3.2)$$

where $dN(t)$ is a point process, which represents the arrivals of packets. For every point at time t , one find a random variable, and the complete form of X_n with variable t is a random process.

The main statistical measures of these processes are

- Mean:

$$\mu = E[X_n] \quad (3.3)$$

- Dispersion:

$$\sigma^2 = \text{Var}[X_n] = E[(X_n - \mu)^2] \quad (3.4)$$

- Covariance:

$$\text{Cov}(X_n, X_{n+k}) = E[(X_n - \mu)(X_{n+k} - \mu)] \quad (3.5)$$

- (Normalized) autocorrelation:

$$R(k, T_i) = \frac{\text{Cov}(X_n, X_{n+k})}{\sigma^2} \quad (3.6)$$

- Dispersion index of counting:

$$F(T) = \frac{\text{Var}[N(T)]}{E[N(T)]} \quad (3.7)$$

- Density of power spectrum:

$$S(\omega) = \sum_{k=-\infty}^{\infty} R(k) e^{-j2k\omega} \quad (3.8)$$

where $\omega = 2\pi f$ is the angular frequency presented in radians per second.

- The continuous component of the power spectrum:

$$S(0) = \sum_{k=-\infty}^{\infty} R(k) \quad (3.9)$$

A random process X_n is considered to be stationary if its statistical properties do not change with time. This means that the process lacks both quantitative and qualitative scale. The first-order statistics are the same on short term as well as on long term. When both first- and second-order statistics (μ and σ^2) are invariant, the random process is considered to be wide-sense stationary or second-order stationary. For such a process, the mean μ is a constant, the dispersion σ^2 is finite, and the autocorrelation function $R(k, T_i)$ depends only on the time difference k . Moreover, if X_1, \dots, X_n are noncorrelated, then $R(k, T_i) = 0$ for $n \neq n + k$, that is, for $k \neq 0$.

3.2.2 Continuous time self-similar processes

A stochastic process is termed fractal when a representative series of statistics presents specific scaling exponents. Because the scaling procedure is mathematically expressed by a power law, one can consider that the network traffic has fractal properties when several statistic estimations display a power-law behavior on a large range of time and frequency scales. A time-continuous stochastic process $X(t)$ is defined as statistical self-similar, with parameter $H(0.5 \leq H \leq 1.0)$ if for any a real and positive,

the processes $X(t)$ and $aHX(at)$ have identical finite dimensional distributions (i.e., with the same statistic properties) for all n positive integers:

$$\{X(t_1), X(t_2), \dots, X(t_n)\} \underset{D}{=} \{a^{-H}X(at_1), a^{-H}X(at_2), \dots, a^{-H}X(at_n)\} \quad (3.10)$$

The term $\underset{D}{=}$ means “asymptotically equal with” from the distributions point of view. Practically, statistical self-similarity implies the validity of the following conditions:

- Mean:

$$E[X(t)] = \frac{E[X(at)]}{a^H} \quad (3.11)$$

- Dispersion:

$$\text{Var}[X(t)] = \frac{\text{Var}[X(at)]}{a^{2H}} \quad (3.12)$$

- Autocorrelation:

$$R(t, \tau) = \frac{R(at, a\tau)}{a^{2H}} \quad (3.13)$$

The Hurst parameter (H) represents a self-similarity degree, that is, the degree of persistence of the statistical phenomenon. The value $H=0.5$ shows the lack of self-similarity, while high values of H (close to 1) shows a high self-similarity degree (or LRD) of a process. In other words, for such a process, an ascendant (or descendent) trend in the past implies, with high probability, an ascendant (or descendent) trend in the future.

3.2.3 Discrete time self-similar processes

Let consider the time series $X = X_n$, $n \in \mathbb{Z}^+$ and the corresponding (m -aggregate) time series $X^{(m)} = X_n^{(m)}$, $n \in \mathbb{Z}^+$ obtained by the average of over m -dimensional nonoverlapping adjacent blocks of the original (basic) time series:

$$X_n^{(m)} = \frac{1}{m} \sum_{i=nm-(m-1)}^{nm} X_i \quad (3.14)$$

$X^{(1)}$ represents the best resolution for the process in this situation. Evolutions with lower resolution of the process $X^{(m)}$ can be obtained by m -aggregation of the process X_n , such as (for $m = 4$):

$$X_n^{(4)} = \frac{X_{4n-3} + X_{4n-2} + X_{4n-1} + X_{4n}}{4} \quad (3.15)$$

The process $X^{(m)}$ represents a less detailed copy of the process $X^{(1)}$. If the statistical properties (mean, dispersion) do not change by aggregation, then the process has self-similar nature. The process $X_n^{(m)}$ can be considered also as a time average of the process X_n . For a stationary ergodic process X_n , the temporal average is equal with the mean of the all set of samples:

$$\bar{X} = E[X_n] = \frac{\sum_{i=1}^n X_i}{n} \quad (3.16)$$

The dispersion of the mean should be equal to the dispersion of the basic random variable divided to the number of samples:

$$\text{Var}(\bar{X}) = \frac{\sigma^2}{n} \quad (3.17)$$

However, due to the persistence of the statistical properties on various timescales, the m -aggregation in a self-similar process is different from the average of all samples of length m , that is, the dispersion approaches zero slower than in the case of a stationary ergodic process:

$$\text{Var}(\bar{X}) \approx \frac{\sigma^2[1 + \delta_n(R)]}{n} \quad (3.18)$$

where

$$\delta_n(R) = \frac{\sum_{i \neq j} R(i, j)}{n} \quad (3.19)$$

and the correlation function of X_i and X_j is

$$R(i, j) = \frac{\text{Cov}(X_i, X_j)}{\sigma^2} \quad (3.20)$$

There are two categories of self-similar processes: exactly self-similar processes and asymptotically self-similar processes.

The process X is exactly self-similar with parameter β ($0 < \beta < 1$) if, for $m \in \mathbb{Z}^+$, the following conditions hold:

- Dispersion

$$\text{Var}[X^{(m)}] = \frac{\text{Var}[X]}{m^\beta} \quad (3.21)$$

- Autocorrelation

$$R(k, X^{(m)}) = R(k, X) \quad (3.22)$$

The parameter β is related with Hurst parameter by

$$\beta = 2(1 - H) \quad (3.23)$$

One can observe that $\beta = 1$ for stationary ergodic process and the dispersion approaches zero.

The process X is asymptotically self-similar if, for big values of k the dispersion is

$$\text{Var}[X^{(m)}] = \frac{\text{Var}[X]}{m^\beta} \quad (3.24)$$

and the autocorrelation function is

$$R(k, X^{(m)}) \rightarrow R(k, X) \quad (3.25)$$

when $m \rightarrow \infty$.

For both categories of self-similar processes, the dispersion of $X^{(m)}$ decays slower than $1/m$ when $m \rightarrow \infty$, in comparison with the stochastic processes where the dispersion decays proportional with $1/m$ and approaches zero when $m \rightarrow \infty$ (like in the case of the white noise, which has a uniform power spectrum). Anyway, the most important characteristic of the self-similar processes is to maintain the autocorrelation when $m \rightarrow \infty$.

3.2.4 Properties of the fractal processes

Fractal processes are closely related to the self-similar processes, exhibiting common characteristics such as LRD, slowly decaying dispersion,

heavy tail distributions, and so on. The most common self-similar process with fractal properties is the fractional Brownian motion (*fBm*).

When a self-similar process $X(t)$ has the property of stationary increments (i.e., when the finite dimensional distributions of $X[t + \tau] - X[t]$ are independent of t), it can be considered a basic fractal process with LRD, slow decaying dispersion and $1/f$ noise aspect:

$$X_n = X[nT_i] - X[(n-1)T_i] \quad (3.26)$$

3.2.4.1 Long-range dependence

The process X is long-range dependent if, for every fixed T_i , the autocorrelation function is not summable:

$$\sum_{k=0}^{\infty} \text{Cov}(k) = \infty \quad (3.27)$$

A better definition is that a process with LRD has a covariance function with hyperbolic decay:

$$\text{Cov}(X_n, X_{n+k}) \sim |k|^{-\beta} \quad (3.28)$$

when $|k| \rightarrow \infty$ and $0 < \beta < 1$ (the definition of parameter β is given by Equation 3.23).

On the contrary, for the SRD processes, the covariance function has an exponential decay:

$$\text{Cov}(X_n, X_{n+k}) \sim a^{|k|} \quad (3.29)$$

when $|k| \rightarrow \infty$ and $0 < a < 1$. Therefore, a SRD process has a summable autocorrelation function

$$\sum_{k=0}^{\infty} \text{Cov}(k) = \text{finite} < \infty \quad (3.30)$$

We can conclude that $\text{Cov}(k)$ becomes small for big values of k , while their cumulative effect is significantly more important in the case of the LRD processes as the one shown in [Figure 3.1](#).

3.2.4.2 Slowly decaying dispersion

As was already mentioned, the dispersion of the samples average decays slower in the case of self-similar processes for m big enough:

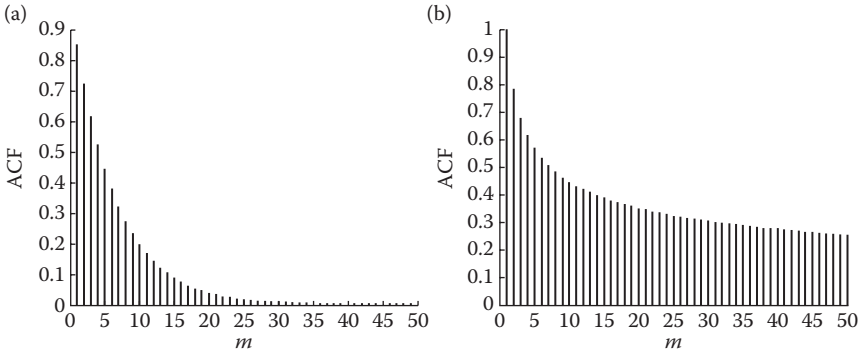


Figure 3.1 Autocorrelation structures for SRD processes (a) and for LRD processes (b).

$$\text{Var}[X^{(m)}] \sim m^{-\beta} \quad (3.31)$$

On the other hand, for SRD processes $\beta = 1$ and

$$\text{Var}[X^{(m)}] \sim m^{-1} \quad (3.32)$$

The property of slowly decaying dispersion can be easily shown by the graphic plot of $\text{Var}[X^{(m)}]$ as function of m on a logarithmic axis (dispersion–time diagram). A straight line with negative slope <1 (on a large range of values of m) points to a slowly decaying dispersion. The same property can be expressed by the dispersion index of counting $F(T)$ (given in Equation 3.7).

The equation

$$\text{Var}[X^{(m)}] = \frac{E[N(mT_i)]F[mT_i]}{m^2} \quad (3.33)$$

shows the equivalence between a dispersion–time diagram, which indicates a slowly decaying dispersion with parameter β and a plot of the dispersion index with slope $1 - \beta$.

3.2.4.3 Heavy-tailed distributions

If the abovementioned properties (covariance and autocorrelation) refer the scaling properties of time-dependent statistics, the attributes of the heavy-tailed distributions are focussed on the marginal amplitudes distribution of X_m , as defined in Equation 3.1. Packets arrivals or ON/OFF intervals are examples of such processes.

A random variable X is called heavy-tailed if the complementary cumulative distribution function (CCDF) decays as a power law:

$$\Pr P\{X \geq x\} \sim x^{-\alpha} \quad (3.34)$$

when $x \rightarrow \infty$ for $\alpha > 0$. For $0 < \alpha \leq 1$ all the moments of X are infinite and in a wide sense, the n th moment is infinite for $n \geq \alpha$. The most known type of heavy-tailed distribution is the Pareto distribution. Let us also mention that although the property of heavy-tailed distribution is not a necessary condition for self-similarity, the self-similar nature of many traffic data results from heavy-tailed distributions (the so called Joseph effect, which illustrates the idea that movements in a time series tend to be part of larger trends and cycles more often than they are completely random; the Joseph effect is quantified by the Hurst component, where movements fall between a Hurst range of 0 and 1).

3.2.4.4 $1/f$ noise

This property is an aspect of the LRD in the frequency domain, more precisely the evidence of the fact that the density of the power spectrum (PSD—defined in Equation 3.8) of the LRD processes diverges at small frequencies, as a contrary in comparison with the SRD processes, which have a smooth PSD at small frequencies, that is,

$$S(\omega) \sim \frac{1}{|\omega|^\gamma} \quad (3.35)$$

when $|\omega| \rightarrow 0$ and $0 < \gamma < 1$. The relation between the parameters γ and β is

$$\gamma = 1 - \beta = 2H - 1 \quad (3.36)$$

For SRD processes, which have a finite PSD when $|\omega| \rightarrow 0$, (i.e., when $\gamma = 0$), the values of the autocorrelation function $R(k)$ decays fast for big values of k , reaching a finite sum.

3.2.4.5 Fractal dimension

The fractal dimension d of an object is defined by

$$d = \frac{\ln N}{\ln(1/\eta)} \quad (3.37)$$

where N is the number of self-similar pieces that cover an d -dimensional object, and η is the dimension of the scale unit. For SRD processes, $d = 1$, while for LRD processes, d has fractioned dimension of form $d = xx.yy....$

3.2.4.6 SRD versus LRD: Conclusion

For LRD processes, the following characteristics are representative:

- $E[X_n^m]$ is close to the second-order moment when $m \rightarrow \infty$.
- $\text{Var}[X^m]$ approaches asymptotically $\text{Var}[X]/m$ when $m \rightarrow \infty$.
- $\sum_{k=0}^{\infty} \text{Cov}(X_n, X_{n+k})$ is convergent.
- The spectrum $S(\omega)$ is finite for $\omega = 0$.

For LRD processes, the following characteristics are representative:

- $E[X_n^m]$ differs the second-order moment when $m \rightarrow \infty$.
- $\text{Var}[X^m]$ approaches asymptotically $m^{-\beta}$ when $m \rightarrow \infty$.
- $\sum_{k=0}^{\infty} \text{Cov}(X_n, X_{n+k})$ is divergent.
- The spectrum $S(\omega)$ is singular for $\omega = 0$.

3.3 Self-similar traffic modeling

We have already shown in Chapter 1 that traffic models are mainly used in traffic engineering to anticipate the performance of the traffic network and to evaluate the ways of improving it. The accuracy of a traffic model consists in the ability to model various correlation structures and marginal distributions. If a model is not able to capture the statistical characteristics of the real traffic, the network will have a poor performance, which is either underestimated or overestimated. The models presented in Chapter 1 were classified as stationary and nonstationary. In turn, stationary models are classified into two categories: short-range dependent (which includes traditional models like Markov processes and autoregressive models—AR, ARMA, ARIMA, TES, and DAR) or long-range-dependent models (fractional ARIMA, fractional Brownian motion). Traditional traffic models have an autocorrelational structure with exponential decay, so $\sum_k \rho_k < \infty$. In this case, the dispersion of the mean v_m decays in a similar manner as m^{-1} for large values of m , while self-similar traffic models have an autocorrelational structure with slower decay, so $\sum_k \rho_k \rightarrow \infty$.

In this chapter, we focus on the modeling of the self-similar characteristics, corresponding to their utilization. These models can be separated into two categories: single-source models and aggregated models. The well-known model will be analyzed in the following.

3.3.1 Single-source traffic models

Internet services are constructed usually after the client–server paradigm in order to transfer information blocks (like files or written messages)

whose dimensions are better characterized by heavy-tailed distributions and eventually lead to long-range-dependent network traffic behavior. Among these distributions, the most usual in modeling sources with high variability are Pareto distribution and log-normal distribution.

3.3.1.1 Pareto distribution

Pareto distribution is a power-law function parameterized by two parameters: the shape parameter α (named also Pareto index) and the scale parameter β (named also cutoff parameter because it represents the minimum value of the random variable X). The cumulative distribution function (CDF) Pareto is given by

$$F(x) = 1 - \left(\frac{\beta}{x}\right)^\alpha \quad (3.38)$$

and the probability density function for $x > \beta$ and $\alpha > 0$

$$f(x) = \frac{\alpha}{\beta} \left(\frac{\beta}{x}\right)^{\alpha+1} \quad (3.39)$$

for $x \leq \beta$

$$f(x) = F(x) = 0 \quad (3.40)$$

The mean (the expected value) is

$$E[X] = \frac{\alpha\beta}{\alpha-1} \quad \text{with } \alpha > 1 \quad (3.41)$$

The dispersion is

$$\sigma^2 = \frac{\alpha}{(\alpha-1)^2(\alpha-2)} \quad (3.42)$$

with $\alpha > 2$ and $\beta = 1$. The parameter α determines the mean and the dispersion as follows:

- For $\alpha \leq 1$, the distribution has infinite mean.
- For $1 \leq \alpha \leq 2$, the distribution has infinite mean and infinite dispersion.
- For $\alpha \leq 2$, the distribution has infinite dispersion.

The relation between parameter α and Hurst parameter is

$$H = \frac{3 - \alpha}{2} \quad (3.43)$$

3.3.1.2 Log-normal distribution

A log-normal distribution is a probability distribution of a random variable whose logarithm is normally distributed. If the stochastic variable $Y = \log X$ is normally distributed, then the variable X is log-normally distributed. The distribution has two parameters, the mean μ of $\ln(x)$, called also location parameter ($\mu > 0$) and the standard deviation σ of $\ln(x)$, called also the scaling parameter ($\sigma > 0$), for $0 \leq x \leq \infty$. The cumulative distribution function and the probability density function are

$$F(x) = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{\ln x - \mu}{\sigma \sqrt{2}} \right) \right] \quad (3.44)$$

$$f(x) = \frac{1}{\sigma x \sqrt{2\pi}} e^{-(\ln x - \mu)^2 / 2\sigma^2} \quad (3.45)$$

The error function $\operatorname{erf}(y)$ is defined as

$$\operatorname{erf}(y) = \frac{2}{\sqrt{\pi}} \int_0^y e^{-t^2} dt \quad (3.46)$$

The mean and the dispersion of the log-normal distribution are

$$E[X] = e^{\mu + (\sigma^2/2)} \quad (3.47)$$

$$\operatorname{Var}[X] = e^{2\mu + \sigma^2} (e^{\sigma^2} - 1) \quad (3.48)$$

The log-normal distributions are useful in many applications such as regressive modeling and the analysis of experimental models, especially because it can model errors produced by many factors.

3.3.2 Aggregate traffic models

Most traffic analysis and modeling studies to date have attempted to understand aggregate traffic, in which all simultaneously active connections are lumped together into a single flow. Typical aggregate time series

include the number of packets or bytes per time unit over some interval. Numerous studies have found that aggregate traffic exhibits fractal or self-similar scaling behavior, that is, the traffic “looks statistically similar” on all timescales. Self-similarity endows traffic with LRD. Numerous studies have also shown that traffic can be extremely bursty, resulting in a non-Gaussian marginal distribution. These findings are in sharp contrast to classical traffic models such as Markov or homogeneous Poisson. LRD and non-Gaussianity can lead to much higher packet losses than predicted by classical Markov/Poisson queuing analyses.

3.3.2.1 ON–OFF models

This approach was first suggested by Mandelbrot, for economical matters, based on renewal processes and later was applied by W. Willinger, M. Taqqu, R. Sherman, and D. Wilson for network communication traffic (Willinger et al. 1997).

The aggregate traffic is generated by multiplexing several independent ON–OFF sources, with each source changing these states in a given periodicity. The generated traffic is (asymptotically) self-similar due to the superposition of several renewal processes, where the renewal values are only 0 and 1, and the interval between renewal is heavy-tailed (e.g., a Pareto distribution with $1 < \alpha < 2$). This superposition presents the syndrome of the infinite dispersion (the so-called Noah effect) and leads to closing of the fractional Brownian motion (fBm) in self-similar aggregate traffic when the number of sources is very high (the so-called Joseph effect), with the Hurst parameter being

$$H = \frac{3 - \alpha}{2} \quad (3.49)$$

The ON–OFF model is simple and therefore very attractive for the simulation of Ethernet traffic, considering each station connected in an Ethernet network as a renewal process, staying in state ON when communicating and in the state OFF when not communicating. So Ethernet traffic is the result of aggregation of several renewal processes.

3.3.2.2 Fractional Brownian motion

The concept of fractional Brownian motion (fBm) was introduced by Mandelbrot and Van Ness in 1968 like a generalization of the Brownian motion (Bm). The $Bm(t)$ process has the property that any random vector at different times has a joint distribution identical to that of a rescaled and normalized version of the random vector, which is a unique continuous Gaussian process with this property and with stationary increments. The (standard) fBm family of processes $B_H(t)$ is indexed by a single parameter $H \in [0, 1]$, which controls all of its properties. Larger values of H

correspond to stronger large-scale correlations, which make the sample paths appear more and more smooth. In contrast, $B_{1/2}$ is the Brownian motion without memory, which in discrete time is just a random walk. The increments of fBm with $H \in [0, 1]$ define the well-known fractional Gaussian noise (fGn) family of stationary processes. For Brownian motion, these increments are i.i.d., which is clearly SRD. fGn processes with $H \in [0, 1/2)$ are also SRD, though qualitatively different from white noise, whereas fGn processes with $H \in (1/2, 1]$ are LRD. Of course, this appearance can be modified by allowing for two additional parameters, namely mean and variance, which allows capturing a wide range of “smoothness” scenarios, from “highly bursty” to “very smooth.” For analytical studies, the canonical process for modeling LRD traffic rate processes is fGn with $H \in (1/2, 1]$. When describing the observed self-similar behavior of measured traffic rate, we can be more flexible. The standard model is a self-similar process in the asymptotic sense or, equivalently, an LRD process.

The following definitions are specific for a fBm process:

The expected value of the process is zero

$$E[B(t + \tau) - B(t)] = 0 \quad (3.50)$$

The dispersion is proportional to the time difference

$$\text{Var}[B(t + \tau) - B(t)] = \sigma^2 |\tau| \quad (3.51)$$

The fGn process $B_H(t)$ is defined as the moving average of $dB(t)$, where the past increments of $B(t)$ are mediated by the kernel $(t - \tau)^H - (1/2)$ and can be expressed by the integral fractional Weyl integral of $B(t)$:

$$B_H(t) = \frac{1}{\Gamma(H + (1/2))} \int_{-\infty}^0 [(t - \tau)^{H-(1/2)} - (-\tau)^{H-(1/2)}] dB(\tau) + \int_0^t (t - \tau)^{H-(1/2)} dB(\tau) \quad (3.52)$$

where $\Gamma(x)$ is the Gamma function. The value of a fBm process at moment t is

$$B_H(t) = Xt^H \quad (3.53)$$

where X is a random variable normally distributed with zero mean and dispersion equal to 1, where due to the self-similarity

$$B_H(at) = a^H B_H(t) \quad (3.54)$$

For $H = 0.5$, one obtains the ordinary Brownian motion (Bm).

To conclude, let consider the main statistical properties of a fBm process

- Mean: $E[B_H(t)] = 0$
- Dispersion: $\text{Var}[B_H(t)] = \text{Var}[X^{t^H}] = t^{2H}$
- Autocorrelation: $R_{B_H}(t, \tau) = E[B_H(t)B_H(\tau)] = 1/2(t^{2H} + \tau^{2H} - |t - \tau|^{2H})$
- Stationary increments: $\text{Var}[B_H(t) - B_H(\tau)] = |t - \tau|^2$

3.3.2.3 Fractional Gaussian noise

The stationary process of increment X_H of fBm type is known as fractional Gaussian noise (fGn):

$$X_H = (X_H(t) = B_H(t+1) - B_H(t) : t \geq 0) \quad (3.55)$$

It is a process with Gaussian distribution, with mean μ and dispersion σ^2 . The dispersion of the mean of the square of increments is proportional to the time difference, and the autocorrelation function is

$$R(t) = \frac{1}{2} (|t+1|^{2H} - |t|^{2H} + |t+1|^{2H}) \quad (3.56)$$

In the limit case when $t \rightarrow \infty$

$$R(t) \sim H(2H-1) |t|^{2H-2} \quad (3.57)$$

A fGn process is exactly self-similar when $0.5 < H < 1$. When $\mu_X = 0$, fGn becomes Gaussian time-continuous process, $B_H = \{B_H(t)\}_{t=0}^\infty$, with $0 < H < 1$ and the autocorrelation function $\mu_X(s, t) = 1/2(|s|^{2H} + |t|^{2H} - |t-s|^{2H})$

3.3.2.4 Fractional ARIMA processes (FARIMA)

An integrated autoregressive moving-average (p, d, q) series, denoted ARIMA (p, d, q), is closely related to ARMA (p, q). ARIMA models are more general as ARMA models and include some nonstationary series (Box and Jenkins 1976).

FARIMA processes are the natural generalizations of standard ARIMA (p, d, q) processes when the degree of differencing d is allowed to take nonintegral values. The mathematical aspects were discussed in detail in Section 2.5.5.

Let remember that for $0 < d < 0.5$, FARIMA (p, d, q) process is asymptotically self-similar with $H = d + 0.5$. An important advantage of this model is the possibility to capture both short- and long-range-dependent correlation structures in time series modeling. The main drawback is the

increased complexity, but it can be reduced using the simplified FARIMA model $(0, d, 0)$.

Consequently, compared with ARMA and fGn processes, the FARIMA (p, d, q) processes are flexible and parsimonious with regard to the simultaneous modeling of the long- and short-range-dependent behavior of a time series.

3.3.2.5 Chaotic deterministic maps

A chaotic map $f:V \rightarrow V$ can be defined by the following three criteria (Erramilli et al. 2002):

1. It shows sensitive dependence on initial conditions (SIC), which means that typical trajectories starting from arbitrarily close initial values nevertheless diverge at an exponential rate. SIC is the basis of the “deterministic” property of chaotic systems, because errors in the estimates of initial states become amplified and prevent accurate prediction of the trajectory’s evolution.
2. It is topologically transitive or strongly mixing, which implies that it is irreducible and that it cannot be decomposed into subsets that remain disjoint under repeated action of the map.
3. Periodic points are dense in V .

From a modeling perspective, even low-order chaotic maps can capture complex dynamic behavior. It is possible to model traffic sources that generate fluctuations over a wide range of timescales using chaotic maps that require a small number of parameters.

If one considers a chaotic map defined as $X_{n+1} = f(X_n)$ and two trends (evolutions) with almost identical conditions X_0 and $X_0 + \epsilon$, then SIC can be expressed by

$$|f^N(X_0 + \epsilon) - f^N(X_0)| = \epsilon e^{N\lambda(x_0)} \quad (3.58)$$

where $f^N(X_0)$ refers to a N time-iterated map, and the parameter $\lambda(x_0)$ describes the exponential divergence (the so-called Liapunov exponent). For a chaotic map, $\lambda(x_0)$ should be positive for “almost all” X_0 . In other words, the limits in the accuracy of the specification of the initial conditions lead to an exponential growth of the incertitude, making the traffic behavior on long-term unpredictable. Another important characteristic is the convergence of the trajectories (in the phase space) to a certain object named strange attractor, which is typical for a fractal structure.

In order to demonstrate how we can associate packet source activity with the evolution of a chaotic map, let consider the one-dimensional map in Figure 3.2 where the state variable x evolves according to $f_1(x)$ in the interval $[0, d)$ and according to $f_2(x)$ in the interval $[d, 1]$.

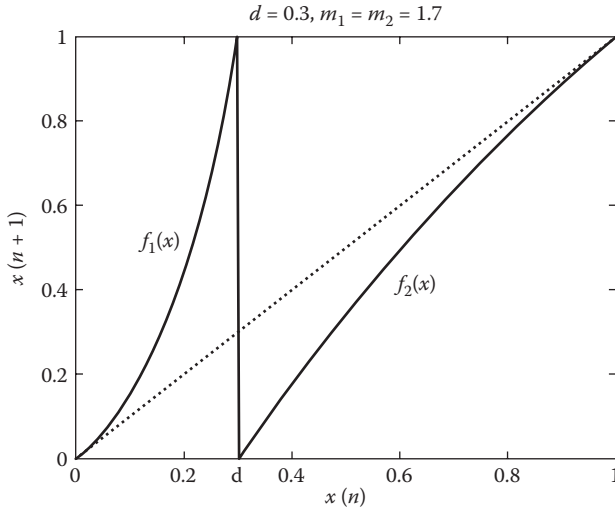


Figure 3.2 Evolutions in a chaotic map.

One can model an on-off traffic source by stipulating that the source is active and generates one or more packets whenever x is in the interval $[d, 1]$, and is idle otherwise. By careful selection of the functions $f_1(x)$ and $f_2(x)$, one can generate a range of on-off source behavior. The traffic generated in this manner can be related to fundamental properties of the map. As the map evolves over time, the probability that is in a given neighborhood is given by its invariant density $\rho(x)$, which is the solution of the Frobenius–Perron equation:

$$\rho(x) = \int_0^1 \delta(x - f(z)) \rho(z) dz \tag{3.59}$$

If the source is assumed to generate a single packet with every iteration in the on-state ($d \leq x \leq 1$), the average traffic rate is simply given by

$$\lambda = \int_0^1 \rho(x) dx \tag{3.60}$$

Another key source characteristic is the “sojourn” or run times spent in the on- and off-states. Given the deterministic nature of the mapping, the sojourn time in any given state is solely determined by the reinjection or initial point x_0 at which the map reenters that state. For example, let

the map reenter the on-state at $x_0 \leq d$. The sojourn time there is then the number of iterations it takes for the map to leave the interval $[d, 1]$:

$$f_2^{k-1}(x_0) \leq d, \quad f_2^k(x_0) < d \quad (3.61)$$

One can then derive the distributions of the on- and off-periods from the reinjection density $\psi(x)$ (where the probability that the map reenters a state in the neighborhood $[x_0, x_0 + dx]$ is given by $\psi[x_0]dx$) and Equation 3.61. The reinjection density can be derived from the invariant density

$$\psi(x) = \rho(f^{-1}(x)) \frac{df^{-1}(x)}{dx} \quad (3.62)$$

One can use these relations to demonstrate the following.

1. The sojourn times in the two states are geometrically distributed if $f_1(x)$ and $f_2(x)$ are linear:

$$f_1(x) = \frac{x}{d}, \quad 0 \leq x < d; \quad f_2(x) = \frac{1-x}{1-d}, \quad d \leq x \leq 1 \quad (3.63)$$

This follows from the fact that the invariant density in this case is uniform, and substituting for the linear mappings in Equation 3.60.

2. In order to match the heavy-tailed sojourn time distribution behavior observed in measurement studies, one can choose $f_1(x)$ such that as $x \rightarrow 0$, $f_1(x) \sim x + cx^m$, where $1.5 < m < 2$. Note that this function evolves very slowly in the neighborhood of 0 (a fixed point). This results in sojourn times in the off-state that are characterized by distributions with infinite variance.

One can similarly generate extended on-times by choosing $f_2(x)$ so that it behaves in a similar manner in the neighborhood of 1 (another fixed point). In this way, one can match the heavy-tailed sojourn time distribution observed in measurement studies.

Thus, chaotic maps can capture many of the features observed with actual packet traffic data, with a relatively small number of parameters. While the qualitative and quantitative modeling of actual traffic using chaotic maps is itself the subject of ongoing research, ultimately the usefulness of this approach will be determined by the feasibility of solving for the 1-D and 2-D invariant densities. We have demonstrated the potential of chaotic maps as models of packet traffic. Given that packet traffic is very irregular and bursty, it is nevertheless possible to construct

simple, low-order nonlinear models that capture much of the complexity. While chaotic maps allow concise descriptions of complex packet traffic phenomena, there are considerable analytical difficulties in their application.

As we indicated earlier, a key aspect of this modeling approach is to devise chaotic mappings $f(x)$ such that the generated arrival process matches measured actual traffic characteristics, while maintaining analytical tractability. Other source interpretations relating the state variable to packet generation are possible and may be more suitable for some applications.

Performance analysis, that is, analysis of queuing systems in which the randomness in the arrival and/or service characteristics is modeled using chaotic maps, particularly systems without a conventional queuing analog requires additional work on the modeling and representation of systems of practical interest using chaotic maps, as well as investigation of numerical and analytical techniques to calculate performance measures of interest, including transient analysis.

3.3.3 Procedures for synthetic self-similar traffic generation

The search for accurate mathematical models of data streams in modern telecommunication networks has attracted a considerable amount of interest; 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, especially when they are engaged in sophisticated Web services. The properties of teletraffic in such scenarios are very different from both the properties of the traditional models of data traffic generated by computers, and drawbacks 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.

Several methods for generating pseudo-random self-similar sequences have been proposed. They include methods based on fast fractional Gaussian noise, fractional ARIMA processes, the $M/G/1$ queue model, autoregressive processes, spatial renewal processes, etc. Some of them generate asymptotically self-similar sequences and require large amounts of CPU time. Even though exact methods of generation of self-similar sequences exist, they are usually inappropriate for generating long sequences because they require multiple passes along the generated sequences. To overcome this problem, approximate methods for generation of self-similar sequences in simulation studies of telecommunication networks have also been proposed. The most popular studies are presented in the following.

3.3.3.1 ON-OFF method

This method (also called aggregation method) was described in [Section 3.3.2](#). It consists in the aggregation and superposition of renewal rewards process (ON/OFF), in which activity (ON) and inactivity (OFF) periods follow a heavy-tailed PDF. We also know that there are some drawbacks associated with the deployment of this technique in the network simulation procedures, such as pitfalls in choosing timescales of interest and number of samples. Despite these issues, this approach could allow an immediate use of widespread network simulation tools, such as Network Simulator 2 – ns2 or the software family from OPNET since there is no need to extend their libraries to support such analytical models.

The main advantage is the possibility to implement the algorithm for parallel processing. The main drawback of the aggregation technique is its large computational requirements per sample produced (because of the large number of aggregated sources). This is the main reason why parallelism is introduced. However, even though the computational requirements are large, its actual computational complexity is $O(n)$ for n produced samples. All the other self-similar traffic generation techniques exhibit poorer complexity than $O(n)$ and they eventually result in a slowdown of the simulation as the simulated time interval increases. As an example, Hosking's method requires $O(n^2)$ computations to generate n numbers (Jeong et al. 1998).

To illustrate the above statements, let consider the following simulation model. Cell arrivals will be represented by Run Length Encoded (RLE) tuples. An RLE tuple t_i includes two attributes, $s(t_i)$, the state of the tuple, and $d(t_i)$, the duration of the tuple. The two attributes represent the discrete time duration $d(t_i)$ over which the state $s(t_i)$ stays the same. The state is either an indication of whether the source is in the ON or OFF state (e.g., 0 for OFF and 1 for ON in a strictly alternating fashion), or the aggregate number of sources active (in the ON state) for the specified duration, that is, for N sources, $s(t_i) \in \{0, 1, \dots, N\}$. Thus, a sequence of t_i 's is sufficient for representing the arrival process from an ON/OFF source or from any arbitrary superposition of such sources. The benefit of such representation is that the activity of the source over several time slots can be encoded as a single RLE tuple. In summary, the algorithm proceeds by generating a large number, N , of individual source traces in RLE form. The utilization of each one of these sources is set to U/N , such that the aggregation of their N results in the desired link utilization to U . Each logical process (LP) of the simulation merges and generates the combined arrival trace for a separate nonoverlapping segment of time, which we call a slice. Thus, each LP is responsible for the generation of the self-similar traffic trace in the form of RLE tuples over a separate segment (slice) of time.

In logical terms, the concatenation of the slices produced by each *LP* in the proper time succession is the desired self-similar process. The *LPs* continue looping generating a different slice each time. The *LP* performs the generation of the self-similar traffic trace by going through the following three steps at each simulated time slice:

1. It generates the merge of the RLE tuple traces of the N individual sources.
2. It aggregates the merged traffic into a link speed equal to the desired access link speed.
3. It corrects the produced RLE trace by incorporating any residual cell counts.

The generation of the individual RLE source traces is also performed in parallel. Each *LP* generates all the slices of a subset of sources that will be necessary to a number of P *LPs* during the generation of the current slices. That is, if P *LPs* are participating in the simulation, each *LP* generates the RLE tuples of P subsequent slices for the sources that it has been assigned to generate. It then sends their $P - 1$ to the other *LPs* for each source it simulates. The individual ON/OFF sources are parameterized accordingly to fit the desired self-similar traffic. Namely:

- a. The shape value, α , of the Pareto distribution used for the ON period is set according to $H = (3 - \alpha)/2$, where H is the desired Hurst value.
- b. Since N ON/OFF sources are aggregated, the per-source utilization of U/N is determined by the ratio of the ON and OFF periods of the individual processes. That is, the average OFF period $E[\text{OFF}]$ is set to $E[\text{OFF}] = E[\text{ON}]$.
- c. The average ON period $E[\text{ON}]$ is set to $E[\text{ON}] = B$, the average burst length, which can be derived from traffic measurements. $E[\text{ON}]$ does not have any impact on the self-similarity, and it can be considered a free variable.

3.3.3.2 Norros method

An exactly self-similar can be generated using a fBm $Z(t)$ process

$$A(t) = mt + \sqrt{am}Z(t) \quad \text{for } -\infty < t < \infty \quad (3.64)$$

$A(t)$ represents the cumulative arrivals process, that is, the total traffic that arrives until the time t . $Z(t)$ is a normalized fBm process with $0.5 < H < 1.0$, where m is the average rate, and a is the shape parameter (the ratio between dispersion and mean in the time unit) (Norros 1995). We call this an (m, a, H) fBm . When this arrival process flows into a buffer

with a deterministic service, the resulting model is what we call $fBm/D/1$. Unfortunately, no simple, closed-form results have been found for this queuing model. However, Norros was able to prove the following approximating formula:

$$P_{tail}(h) \approx \exp\left(-\frac{1}{2am}\left(\frac{C-m}{H}\right)^{2H}\left(\frac{h}{1-H}\right)^{2-2H}\right) \quad (3.65)$$

where C is the speed at which the packets in the buffer are processed. In an infinite buffer (theoretical abstraction), $P_{tail}(h)$ represents the probability for the buffer length to exceed h bytes. (Regarded as a function of the variable h , the right-hand side is a Weibull distribution.)

There are two problems with this formula. From a theoretic standpoint, the approximations made are nonsystematic (i.e., all in the same direction, either “ \leq ” or “ \geq ”), so Equation 3.65 is neither an upper bound nor a lower bound of the actual $P_{tail}(h)$. The exponential could be either smaller or larger than $P_{tail}(h)$ and there is no estimation of the tightness of the approximation. From an empirical standpoint, Equation 3.65 does not fit into any of our data series, most of the time deviating by orders of magnitude. Even assuming that for an exact fBm process Norros’ formula would be a good approximation, we still have to deal with the fact that our time series are not accurately described by fBm at low timescales.

3.3.3.3 $M/G/\infty$ queue method

Let consider a queue $M/G/\infty$ where the arrivals of the clients follow a Poisson distribution and the service time are obtained from a heavy-tailed distribution (with infinite dispersion). Then the process $\{X_t\}$, representing the number of clients in the system at the t moment, is asymptotically self-similar.

To exemplify this method, we present a mechanism for congestion avoidance in a packet-switched environment. If there are no resources reserved for a connection, the possibility of packets arriving at a node at a rate higher than the processing rate then appears.

Packets arriving on any of a number of input ports (A–Z) can be destined for the same output port. Although there could be, in general, many operations to be performed, they are broadly divided into two classes: serialization (i.e., the process of transmitting the bits out on the link) and everything else that happens inside the node (e.g., classification, rewriting of the TTL, and/or TOS fields in the IP header, routing table and access list look-ups, policing, shaping, etc.) called internal processing. Serialization delays are deterministic and determined by the speed of the output link. Internal processing delays are essentially random (a larger routing table requires a longer time to search) but advanced algorithms

and data structures are doing, in general, a good job of maintaining overall deterministic performance. As a consequence, the only truly random part of the delay is the time a packet spends waiting for service.

The internal workings of a node are relatively well understood (by either analysis or direct measurement) and controllable. In contrast, the (incoming) traffic flows are neither completely understood nor easily controllable. The focus is therefore on accurately describing (modeling) these flows and predicting their impact on network performance.

The picture in [Figure 3.3a](#) is a simplified representation of a packet-switching node (router or switch), illustrating the mechanism of congestion for a simple M/D/1 queue model. The notation is easily explained: “M” stands for the (unique) input process, which is assumed Markovian, that is, a Poisson process; “D” stands for the service, assumed deterministic; and “1” means that only one facility is providing service (no parallel service).

It has been shown time and again that the high variability associated with LRD and SS processes can greatly deteriorate network performance, creating, for instance, queuing delays orders of magnitude higher than those predicted by traditional models. We make this point once again with the plot shown in [Figure 3.3b](#): the average queue length created in a queuing simulation by the actual packet trace is widely divergent from the average predicted by the corresponding queuing model (M/D/1).

In view of our simple model of a node discussed earlier, hypotheses “D” and “1” seem reasonable, so we would like to identify “M” as the cause of the massive discrepancy. The question then arises: Is there an “fBm/D/1” queuing model? The answer is “almost,” which suggests that the self-similarity is mainly caused by user/application characteristics (i.e., Poisson arrivals of sessions, highly variable session sizes or durations) and is hence likely to remain a feature of network traffic for some time to come—assuming the way humans tend to organize information will not change drastically in the future.

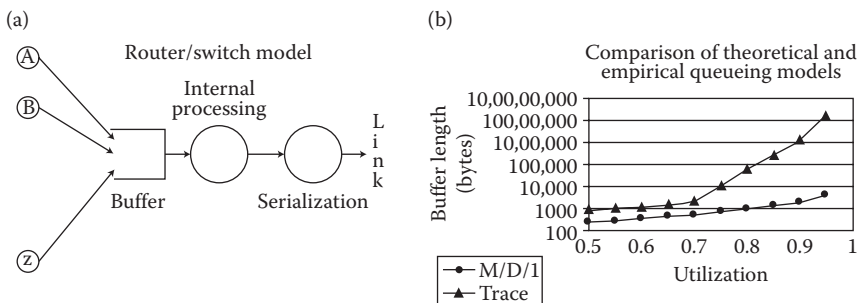


Figure 3.3 A queue model (a) and the comparison of the real traffic with the simulated traffic (b).

3.3.3.4 Random midpoint displacement (RMD) method

The basic concept of the random midpoint displacement (RMD) algorithm is to extend the generated sequence recursively, by adding new values at the midpoints from the values at the endpoints. Figure 3.4 outlines how the RMD algorithm works.

Figure 3.5 illustrates the first three steps of the method, leading to generation of the sequence $(d_{3,1}, d_{3,2}, d_{3,3}, \dots)$ and $d_{3,4}$. The reason for subdividing the interval between 0 and 1 is to construct the Gaussian increments of X . Adding offsets to midpoints makes the marginal distribution of the final result normal.

Step 1. If the process $X(t)$ is to be computed for time instance t between 0 and 1, then start out by setting $X(0) = 0$ and selecting $X(1)$ as a pseudo-random number from a Gaussian distribution with mean 0 and variance $\text{Var}[X(1)] = \sigma_0^2$. Then $\text{Var}[X(1) - X(0)] = \sigma_0^2$.

Step 2. Next, $X(1/2)$ is constructed as the average of $X(0)$ and $X(1)$, that is, $X(1/2) = 1/2 [X(0) + X(1)] + d_1$.

The offset d_1 is a Gaussian random number (GRN), which should be multiplied by a scaling factor $1/2$, with mean 0 and variance S_1^2 of d_1 . For $\text{Var}[X(t_2) - X(t_1)] = |t_2 - t_1|^{2H} \sigma_0^2$ to be true, for $0 \leq t_1 \leq t_2 \leq 1$,

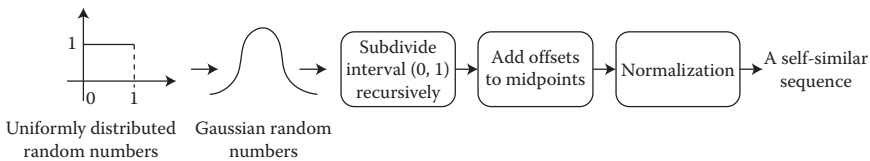


Figure 3.4 Block scheme for the RMD method.

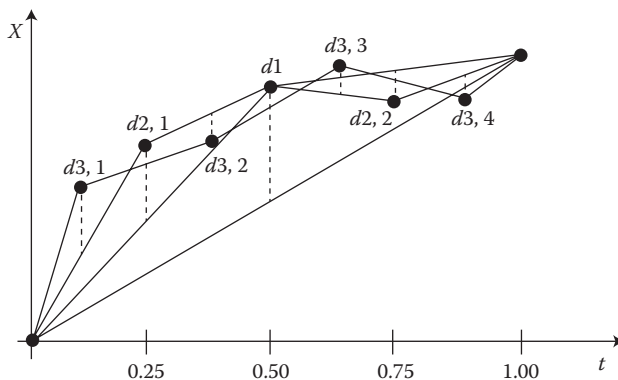


Figure 3.5 The first three steps in the RMD method.

it must be required that $\text{Var}[X(1/2) - X(0)] + S_1^2 = (1/2)^{2H} \sigma_0^2$. Thus, $S_1^2 = (1/2)^{2H} (1 - 2^{2H-2}) \sigma_0^2$

Step 3. Reduce the scaling factor by $\sqrt{2}$, that is, now assume $\sqrt{8}$, and divide the two intervals from 0 and 1/2 and from 1/2 to 1 again. $X(1/4)$ is set as the average $1/2 [X(0) + X(1/2)]$ plus an offset $d_{2,1}$, which is a GRN multiplied by the current scaling factor $\sqrt{1/8}$. The corresponding formula holds for $X(3/4)$, that is, $X(3/4) = 1/2 [X(1/2) + X(1)] + d_{2,2}$ where $d_{2,2}$ is a random offset computed as before. So the variance S_2^2 of $d_{2,*}$ must be chosen such as follows:

$$\text{Var} \left[X \left(\frac{1}{4} \right) - X(0) \right] = \left(\frac{1}{4} \right) \text{Var} \left[X \left(\frac{1}{2} \right) - X(0) \right] + S_2^2 = \left(\frac{1}{2^2} \right)^{2H} \sigma_0^2$$

Thus, $S_2^2 = (1/2^2)^{2H} (1 - 2^{2H-2}) \sigma_0^2$.

Step 4. The fourth step proceeds in the same manner: reduce the scaling factor by, that is, do $\sqrt{16}$. Then set

$$X \left(\frac{1}{8} \right) = \left(\frac{1}{2} \right) \left(X(0) + X \left(\frac{1}{4} \right) \right) + d_{3,1}; \quad X \left(\frac{3}{8} \right) = \left(\frac{1}{2} \right) \left(X \left(\frac{1}{4} \right) + X \left(\frac{1}{2} \right) \right) + d_{3,2}$$

$$X \left(\frac{5}{8} \right) = \left(\frac{1}{2} \right) \left(X \left(\frac{1}{2} \right) + X \left(\frac{3}{4} \right) \right) + d_{3,3}; \quad X \left(\frac{7}{8} \right) = \left(\frac{1}{2} \right) \left(X \left(\frac{3}{4} \right) + X(1) \right) + d_{3,4}$$

In the formulas above, $d_{3,*}$ is computed as a different GRN multiplied by the current scaling factor $\sqrt{16}$. The following step computes $X(t)$ at $t = 1/16, 3/16, \dots, 5/16$ using a scaling factor again reduced by $\sqrt{2}$, and continues as indicated above. So the variance S_3^2 of $d_{3,*}$ must be chosen such that $\text{Var}[X(1/8) - X(0)] = (1/4) \text{Var}[X(1/5) - X(0)] + S_3^2 = (1/2^3)^{2H} \sigma_0^2$.

Thus, $S_3^2 = (1/2^3)^{2H} (1 - 2^{2H-2}) \sigma_0^2$.

The variance S_n^2 of $d_{n,*}$ therefore, yields $S_n^2 = (1/2^n)^{2H} (1 - 2^{2H-2}) \sigma_0^2$.

The main drawback of the method is the generation of a process, which is only approximate self-similar. In particular, the parameter H of the samples trajectories tend to overpass the target value for $0.5 < H < 0.75$ and to be less than the target value for $0.75 < H < 1.0$.

3.3.3.5 Wavelet transform-based method

This method (Abry and Veitch 1998) is based on the so-called multiresolution analysis (MRA), which consists of splitting the sequence into a (low pass) approximation and a series of details (high pass). First, the

coefficients corresponding to the wavelet transform of fBm . Then, these are used in an inverse wavelet transform in order to obtain fBm samples. The disadvantage is that it is only approximate, because generally the wavelet coefficients are not correlated. Nevertheless, the problem can be compensated in the case of wavelets with a sufficient number of moments reduced to zero (i.e., the degree of the polynomial for that the internal scalar product with the wavelet is zero).

For some time after the discovery of scaling in traffic, it was debated as to whether the data was indeed consistent with self-similar scaling, or that the finding was merely an artifact of poor estimators in the face of data polluted with nonstationarities. The introduction of wavelet-based estimation techniques to traffic helped greatly to resolve this question, as they convert temporal LRD to SRD in the domain of the wavelet coefficients, and simultaneously eliminate or reduce certain kinds of nonstationarities. It is now widely accepted that scaling in traffic is real, and wavelet methods have become the method of choice for detailed traffic analysis. Another key advantage of the wavelet approach is its computational complexity (memory complexity is also in an online implementation), which is invaluable for analyzing the enormous data sets of network-related measurements. However, even wavelet methods have their problems when applied to certain real-life or simulated traffic traces. An important rule-of-thumb continues to be to use as many different methods as possible for checking and validating whether or not the data at hand is consistent with any sort of hypothesized scaling behavior, including self-similarity.

Wavelet analysis is a joint timescale analysis. It replaces $X(t)$ by a set of coefficients, such as $d_x(j, k)$, $j, k \in \mathbb{Z}$ where 2^j denotes the scale, and the 2^k instant about which the analysis is performed. In the wavelet domain, Equation 3.57 is replaced by $\text{Var}[d_x(j, k)] = c_f C 2^{(1-\beta)j}$, where the role of m is played by scale, of which j is the logarithm, where c_f is the frequency domain analog of c_ν and C is independent of j . The analog of the variance-time diagram, which is the estimates of $\log(\text{Var}[d_x(j, \cdot)])$ against j , called the logscale diagram. It constitutes an estimate of the spectrum of the process in log-log coordinates, where the low frequencies correspond to large scales, appearing in the right in the figures below. The global behavior of data, as a function of scale, can be efficiently examined in the logscale diagram, and the exponent estimated by a weighted linear regression over the scales where the graph follows a straight line. What constitutes a straight line can be judged using the confidence intervals at each scale, which can be calculated or estimated. The important fact is that the estimation is heavily weighted toward the smaller scales, where there is much more data.

3.3.3.6 Successive random addition (SRA)

Another alternative method for the direct generation of fBm process is based on the successive random addition (SRA) algorithm. The SRA

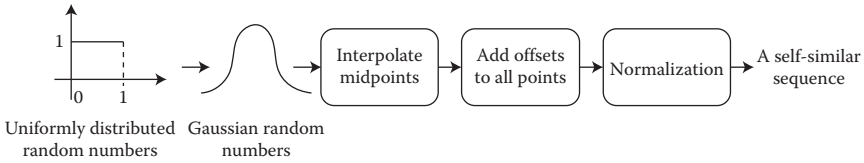


Figure 3.6 Block scheme of the SRA method.

method uses the midpoints like RMD, but adds a displacement of a suitable variance to all of the points to increase stability of the generated sequence.

Figure 3.6 shows how the SRA method generates an approximate self-similar sequence. The reason for interpolating midpoints is to construct Gaussian increments of X , which are correlated. Adding offsets to all points should make the resulted sequence self-similar and of normal distribution.

The SRA method consists of the following four steps:

Step 1. If the process $\{X_i\}$ is to be computed for time instance t between 0 and 1, then start out by setting $X_0 = 0$ and selecting X_1 as a pseudo-random number from a Gaussian distribution with mean 0 and variance $\text{Var}[X_1] = \sigma_0^2$. Then $\text{Var}[X_1 - X_0] = \sigma_0^2$.

Step 2. Next, $X_{1/2}$ is constructed by the interpolation of the midpoint, that is, $X_{1/2} = (1/2)(X_0 + X_1)$.

Step 3. Add a displacement of a suitable variance to all of the points, that is, $X_0 = X_0 + d_{1,1}$, $X_{1/2} = X_{1/2} + d_{1,2}$, and $X_1 = X_1 + d_{1,3}$.

The offsets $d_{1,*}$ are governed by fractional Gaussian noise.

For $\text{Var}[X(t_2) - X(t_1)] = |t_2 - t_1|^{2H} \sigma_0^2$ to be true, for any $t_1, t_2, 0 \leq t_1 \leq t_2 \leq 1$, it must be required that $\text{Var}[X_{1/2} - X_0] = (1/4) \text{Var}[X_1 - X_0] + 2S_1^2 = (1/2)^{2H} \sigma_0^2$.

Thus, $S_1^2 = (1/2)(1/2)^{2H}(1 - 2^{2H-2})\sigma_0^2$.

Step 4. Next, Steps 2 and 3 are repeated. Therefore $S_n^2 = (1/2)(1/2^n)^{2H}(1 - 2^{2H-2})\sigma_0^2$, where σ_0^2 is an initial variance and $0 < H < 1$.

Using the above steps, the SRA method generates an approximate self-similar fBm process.

3.3.4 Fast Fourier transform (FFT) method

A faster method of generating synthetic self-similar traffic implies is the fast Fourier transform (Erdos and Renyi 1960). A complex number sequence is generated so that it corresponds to the power spectrum of a fGn process. Then, the discrete Fourier transform (DTFT) is used to obtain

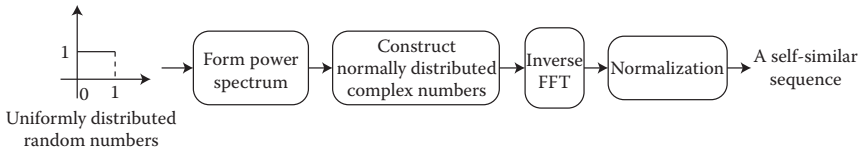


Figure 3.7 Block scheme of the FFT method.

the time correspondent of the power spectrum. Because the sequence has the power spectrum of fGn and the self-correlation and the power spectrum form a Fourier pair, it implies that this sequence has the self-correlation properties of fGn. In addition, DTFT and its inverse can be calculated using the FFT algorithm. The main advantage of this method is its swiftness.

This method generates approximate self-similar sequences based on the FFT and a process known as the fractional Gaussian noise (fGn) process (see Figure 3.7, which shows how the FFT method generates self-similar sequences). Its main difficulty is connected with calculating the power spectrum, which involves an infinite summation.

Briefly, it is based on (i) calculation of the power spectrum using the periodogram (the power spectrum at a given frequency represents an independent exponential random variable); (ii) construction of complex numbers, which are governed by the normal distribution; and (iii) execution of the inverse FFT. This leads to the following algorithm:

Step 1. Generate a sequence of the values $\{f_{1'} \dots, f_{n/2}\}$ where: $\{f_i = \hat{f}(2\pi i/n, H)\}$, corresponding to the power spectrum of a fGn process for frequencies from $2\pi/n$ to π , $0.5 < H < 1$. For a fGn process, the power spectrum $f(\lambda, H)$ is defined as

$$f(\lambda, H) = A(\lambda, H)[|\lambda|^{-2H-1} + B(\lambda, H)] \quad \text{for } 0 < H < 1 \quad \text{and} \quad -\pi \leq \lambda \leq \pi$$

where

$$A(\lambda, H) = 2 \sin(\pi H) \Gamma(2H + 1) (1 - \cos \lambda)$$

$$B(\lambda, H) = \sum_{i=1}^n [(2\pi i + \lambda)^{-2H-1} + (2\pi i - \lambda)^{-2H-1}]$$

Step 2. Adjust the sequence of $\{f_{1'} \dots, f_{n/2}\}$ for estimating power spectrum using periodogram.

Step 3. Generate $\{Z_1, \dots, Z_{n/2}\}$ a sequence of complex values such that $|Z_i| = \sqrt{\hat{f}_i}$ and the phase of Z_i is uniformly distributed between 0 and 2π .

Step 4. Construct $\{Z'_0, \dots, Z'_{n-1}\}$, an expanded version of $\{Z_1, \dots, Z_{n/2}\}$

$$Z'_i = \begin{cases} 0, & \text{if } i = 0 \\ Z_i, & \text{if } 0 < i \leq \frac{n}{2} \\ \bar{Z}_{n-i}, & \text{if } \frac{n}{2} \leq i < n \end{cases}$$

where \bar{Z}_{n-i} denotes the complex conjugate of Z_{n-i} .

$\{Z'_i\}$ retains the power spectrum used in constructing $\{Z_i\}$, but because it is symmetric about $\{Z'_{n/2}\}$, it now corresponds to the FFT of a real-valued signal.

Step 5. Calculate inverse FFT $\{Z'_i\}$ to obtain the approximate fGn sequence $\{X_i\}$.

3.4 Evidence of self-similarity in real traffic

As was already mentioned, the most used indicator for the level of self-similarity in time series is the Hurst parameter H . The self-similarity phenomenon is present for H in the range $0.5 \leq H \leq 1.0$, being stronger for values of H close to 1.0. But the estimation of H is a difficult task, especially because the precise value of H infinite time series should be examined, but actually we dispose of only the finite time series.

There are several methods for the estimation of the self-similarity in a time series, but the most known are statistical R/S (rescaled range) analysis, dispersion–time analysis, periodogram-based analysis, Whittle estimator, and wavelet-based analysis.

3.4.1 Rescaled range method

The rescaled range (R/S) is a normalized, nondimensional measure, proposed by Hurst himself to characterize the data variability. For a given set of experimental observations $X = \{X_n, n \in Z^+\}$ with the sample average $\bar{X}(n)$, sample dispersion $S^2(n)$, and sample range $R(n)$, the rescaled adjusted range (or R/S statistic) is

$$\frac{R(n)}{S(n)} = \frac{\max(0, \Delta_1, \Delta_2, \dots, \Delta_n) - \min(0, \Delta_1, \Delta_2, \dots, \Delta_n)}{S(n)} \quad (3.66)$$

where

$$\Delta_k = \sum_{i=1}^k X_i - k\bar{X} \quad \text{for } k = 1, 2, \dots, n \quad (3.67)$$

In the case of many natural phenomena, when $n \rightarrow \infty$, then

$$E \left[\frac{R(n)}{S(n)} \right] \sim cn^H \quad (3.68)$$

with c a constant positive integer value. Applying logarithms:

$$\log \left\{ E \left[\frac{R(n)}{S(n)} \right] \right\} \sim H \log(n) + \log(c) \quad (3.69)$$

So, one can estimate the value of H by the slope of a straight line that approximate the graphical plot of $\log\{E[R(n)/S(n)]\}$ as a function of $\log(n)$.

The R/S method is not very accurate and it is mainly used only to certify the existence of self-similarity in a time series.

3.4.2 Dispersion–time analysis

The dispersion–time analysis (or variance–time analysis) is based on the property of slowly decrease of the dispersion of a self-similar aggregate process:

$$\text{Var}[X^{(m)}] = \frac{\text{Var}[X]}{m^\alpha} \quad (3.70)$$

where $\alpha = 2 - 2H$.

Equation 3.8 can be written as

$$\log\{\text{Var}[X^{(m)}]\} \sim \log\{\text{Var}(X)\} - \alpha \log(m) \quad (3.71)$$

Because $\log\{\text{Var}[X^{(m)}]\}$ is a constant that not depends on m , one can represent $\text{Var}(X)$ as a function of m in logarithmic axes. The line that approximates the resulting points has the slope α . The values of α between $(-1; 0)$ represent self-similarity.

Like the R/S method, the variance–time analysis is a heuristic method. Both methods can be affected by poor statistics (few realisations of the self-similar process). They offer only a raw estimation of H .

3.4.3 Periodogram method

The estimation based on periodogram is a more accurate method than those based on aggregation, but it necessitates knowing *a priori* the mathematical parameterized model of the process. The periodogram is known also as the function of intensity $I_N(\omega)$ and represents the estimated spectral density of the stochastic process $X(t)$ (defined at discrete time intervals) and given, or the time period N by the equation:

$$I_N(\omega) = \frac{1}{2\pi N} \left| \sum_{k=1}^N X_k e^{jk\omega} \right|^2 \quad (3.72)$$

where ω is the frequency and X_k is the time series. When $\omega \rightarrow 0$ the periodogram should be

$$I_N(\omega) \sim |\omega|^{1-2H} \quad (3.73)$$

The main drawback of this method is the need of a high processing capacity.

3.4.4 Whittle estimator

Whittle estimator derives from the periodogram, but uses a nongraphic representation of the probability. Considering a self-similar process with a *fBm*-type model and its spectral density $S(\omega, H)$, the value of the parameter H minimizes the so-called Whittle expression:

$$\int_{-\pi}^{\pi} \frac{I_N(\omega)}{S(\omega, H)} d\omega \quad (3.74)$$

Both periodogram and Whittle estimator are derived from the maximum likelihood estimation (MLE) and offer good statistical properties, but can give errors if the model of the spectral density is false. Since the self-similarity estimation is concerned only with the LRD in the data sets, the Whittle estimator is employed as follows.

Each hourly data set is aggregated at increasing levels m , and the Whittle estimator is applied to each m -aggregated dataset using the *fGn* model. This approach exploits the property that any long-range-dependent process approaches *fGn* when aggregated to a sufficient level, and so should be coupled with a test of the marginal distribution of the aggregated observations to ensure that it has converged to the normal

distribution. As m increases, SRDs are averaged out of the data set; if the value of H remains relatively constant, we can be confident that it measures a true underlying level of self-similarity. Since aggregating the series shortens it, confidence intervals will tend to grow as the aggregation level increases; however, if the estimates of H appear stable as the aggregation level increases, then we consider the confidence intervals for the unaggregated data set to be representative.

3.4.5 Wavelet-based method

This method, resembling the dispersion–time method, allows the analysis in both time and frequency domains. The wavelet estimator is constructed on the basis of the discrete wavelet transform (DWT), having the cumulative advantages of the estimators based on aggregation and of those based on MLE, but avoiding their drawbacks. The DWT uses the multiresolution analysis (MRA), which consists in the division of the sequence $x(t)$ in a low-pass approximation and several high-pass details, associated with the a_x , respectively, d_x coefficients:

$$x(t) = \text{approx}_J(t) + \sum_{j=1}^J \text{detail}_j(t) = \sum_k a_x(J, k) \varphi_{j,k}(t) + \sum_{j=1}^J \sum_k d_x(j, k) \Psi_{j,k}(t) \quad (3.75)$$

The parameters $\{\varphi_{j,k}(t) = 2^{-j/2} \varphi_0(2^{-j}t - k)\}$ and $\{\Psi_{j,k}(t) = 2^{-j/2} \Psi_0(2^{-j}t - k)\}$ are the sets of the dilated scaling function φ_0 and wavelet function Ψ_0 . The DWT consists in the sets of coefficients $\{d_x(j, k), j = 1, \dots, J, k \in \mathbb{Z}\}$ and $\{a_x(J, k), k \in \mathbb{Z}\}$ defined by the vectorial products:

$$d_x(j, k) = \langle x, \Psi_{j,k} \rangle \quad (3.76)$$

$$a_x(j, k) = \langle x, \varphi_{j,k} \rangle \quad (3.77)$$

These coefficients are placed in a dyadic lattice. The details described by the wavelet coefficients grow when the resolution decrease. In the frequency domain, it corresponds to the effect of a low-pass filter, which produces a relatively constant bandwidth. On the other hand, the spectral estimators (based on periodograms) can be easily influenced, because a constant bandwidth alters the analyzed power spectrum, although it offers a perfect match. Therefore, $\text{Var}[\text{detail}_j] \sim 2^j(2^{H-2})$ and H can be estimated by the linear approximation of a plot in a diagram with logarithmic axes:

$$\log_2 \left(\frac{2^j}{n_0} \sum_k |d_x(j, k)|^2 \right) = (2\hat{H} - 1)j + c \quad (3.78)$$

where \hat{H} is a semiparametric estimator of H , n_0 is the data length, and c is finite. This estimator is efficient in Gaussian hypotheses.

3.5 Application specific models

3.5.1 Internet application-specific traffic models

3.5.1.1 Network traffic

Network traffic is obviously driven by applications. The most common applications that come to mind might include Web, e-mail, peer-to-peer file sharing, and multimedia streaming. A traffic study of Internet backbone traffic showed that Web traffic is the single largest type of traffic, on the average more than 40% of total traffic (Fraleigh et al. 2003). Although on a minority of links, peer-to-peer traffic was the heaviest type of traffic, indicating a growing trend. Streaming traffic (e.g., video and audio) constitute a smaller but stable portion of the overall traffic. E-mail is an even smaller portion of the traffic. Therefore, traffic analysts have attempted to search for models tailored to the most common Internet applications (Web, peer-to-peer, and video).

3.5.1.2 Web traffic

The World Wide Web is a familiar client-server application. Most studies have indicated that Web traffic is the single largest portion of overall Internet traffic, which is not surprising considering that the Web browser has become the preferred user-friendly interface for e-mail, file transfers, remote data processing, commercial transactions, instant messages, multimedia streaming, and other applications.

An early study focused on the measurement of self-similarity in Web traffic (Crovella and Bestavros 1997). Self-similarity had already been found in Ethernet traffic. Based on days of traffic traces from hundreds of modified Mosaic Web browser clients, it was found that Web traffic exhibited self-similarity with the estimated Hurst parameter H in the range 0.7–0.8. More interestingly, it was theorized that Web clients could be characterized as on/off sources with heavy-tailed on periods. In previous studies, it had been demonstrated that self-similar traffic can arise from the aggregation of many on/off flows that have heavy-tailed on or off periods. Heavy tails implies that very large values have nonnegligible probability. For Web clients, on periods represent active transmissions of Web files, which were found to be heavy-tailed. The off periods were either “active

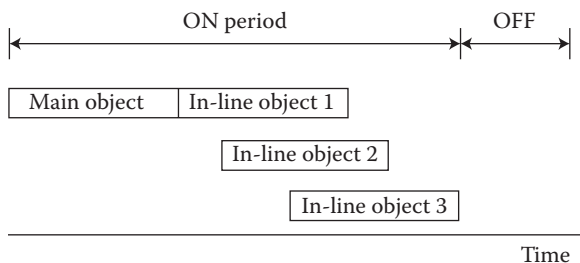


Figure 3.8 ON/OFF model with Web page.

off,” when a user is inspecting the Web browser but not transmitting data, or “inactive off,” when the user is not using the Web browser. Active off periods could be fit with a Weibull probability distribution, while inactive off periods could be fit by a Pareto probability distribution. The Pareto distribution is heavy-tailed. However, the self-similarity in Web traffic was attributed mainly to the heavy-tailed on periods (or essentially, the heavy-tailed distribution of Web files). For example, let consider that a Web page consists of a main object, which is an HTML document, and multiple in-line objects that are linked to the main object (see Figure 3.8). Both main object sizes and in-line object sizes are fit with (different) log-normal probability distributions. The number of in-line objects associated with the same main object follow a gamma probability distribution. In the same Web page, the intervals between the start times of consecutive in-line objects are fit with a gamma probability distribution.

TCP flow models seem to be more natural for Web traffic than “page based” consisting of a main object linked to multiple in-line objects.

In this example, the requests for TCP connections originate from a set of Web clients and go to one of multiple Web servers. A model parameter is the random time between new TCP connection requests. A TCP connection is held between a client and server for the duration of a random number of request–response transactions. Additional model parameters include sizes of request messages, server response delay sizes of response messages, and delay until the next request message. Statistics for these model parameters are collected from days of traffic traces on two high-speed Internet access links.

3.5.1.3 Peer-to-peer traffic

Peer-to-peer (P2P) traffic is often compared with Web traffic because P2P applications work in the opposite way from client–server applications, at least in theory. In actuality, P2P applications may use servers. P2P applications consist of a signaling scheme to query and search for file locations, and a data transfer scheme to exchange data files. While the data transfer

is typically between peer and peer, sometimes the signaling is done in a client-server way. Since P2P applications have emerged only in the last decade, there are a few traffic studies to date. They have understandably focused more on collecting statistics than the harder problem of stochastic modeling. An early study examined the statistics of traffic on a university campus network (Saroiu et al. 2002).

The observed P2P applications were Kazaa and Gnutella. The bulk of P2P traffic was AVI and MPEG video, and a small portion was MP3 audio. With this mix of traffic, the average P2P file was about 4 MB, three orders of magnitude larger than the average Web document. A significant portion of P2P was dominated by a few very large objects (around 700 MB) accessed 10–20 times. In other words, a small number of P2P hosts are responsible for consuming a large portion of the total network bandwidth.

3.5.1.4 Video

The literature on video traffic modeling is vast because the problem is very challenging. The characteristics of compressed video can vary greatly depending on the type of video that determines the amount of motion (from low-motion videoconferencing to high motion broadcast video) and frequency of scene changes; and the specific intraframe and interframe video coding algorithms. Generally, interframe encoding takes the differences between consecutive frames, compensating for the estimated motion of objects in the scene. Hence, more motion means that the video traffic rate is more dynamic, and more frequent scene changes is manifested by more “spikes” in the traffic rate. While many video coding algorithms exist, the greatest interest has been in the Moving Picture Coding Experts Group (MPEG) standards. Video source rates are typically measured frame by frame (which are usually 1/30 s apart). That is, X_n represents the bit rate of the n th video frame. Video source models address at least these statistical characteristics: the probability distribution for the amount of data per frame; the correlation structure between frames; times between scene changes; and the probability distribution for the length of video files. Due to the wide range of video characteristics and available video coding techniques, it has seemed impossible to find a single universal video traffic. The best model varies from video to video file. A summary of research findings was presented by Chen (2007) and includes the following:

- The amount of data per frame has been fit to log-normal, gamma, and Pareto probability distributions or their combinations.
- The autocorrelation function for low-motion video is roughly exponential, while for broadcast video is more complicated, exhibiting both SRD at short lags and LRD at much longer lags.
- Time series, Markov modulated, and wavelet models are popular to capture the autocorrelation structure.

- The lengths of scenes usually follow Pareto, Weibull, or Gamma probability distributions.
- The amount of data in scene change frames are uncorrelated and follow Weibull, Gamma, or unknown probability distributions.
- The lengths of streaming video found on the Web appear to have a long-tailed Pareto probability distribution.

3.5.2 Models for TCP flows

Traffic sources may be limited by access control or the level of network congestion. Some protocols limit the sending rates of hosts by means of sliding windows or closed loop rate control. Clearly, traffic models must take such protocols into account in order to be realistic. Here, we focus on TCP because TCP traffic constitutes the vast majority of Internet traffic. TCP uses a congestion avoidance algorithm that infers the level of network congestion and adjusts the transmission rate of hosts accordingly. Even other protocols should be “TCP friendly,” meaning that they interact fairly with TCP flows.

The transmission control protocol (TCP) has for some time carried the bulk of data in the Internet; for example, currently all Web, FTP (file transfer), and NNTP (network news) services, or some 70%–90% of all traffic, use TCP. For this reason, it is a natural choice of protocol on which to focus attention. TCP aims to provide a reliable delivery service. To ensure this, each packet has a sequence number, and its successful receipt must be signaled by a returning acknowledgment (ACK) packet. The second aim of TCP is to provide efficient delivery. An adaptive window flow control is employed where only a single window of data is allowed to be transmitted at any one time. Once one window’s worth of data has been sent, the sender must wait until at least part of it has been acknowledged before sending more. This method is powerful as it is controlled by the sender, requiring no additional information from either the receiver or the network. TCP has two control windows, the sender-based *cwnd*, and the (advertised) receiver window *rwnd*. Sources use the minimum of the two. Usually we assume that *rwnd* is constant, playing the role of the maximum value of *cwnd*, the dynamical variable of interest, and all packets are of maximum segment size (MSS).

The data send rate r_s of a source with window size w , and packet acknowledgment round-trip time (RTT) is $r_s = w/\text{RTT}$. The interesting design issue in the flow control lies in how to choose *cwnd*. Ideally, the window is exactly tuned to the available bandwidth in the network. If it is too small, then the network is used inefficiently, while if too large, congestion may result. However, the Internet bandwidths can vary from kb/s to Gb/s, while RTTs can range from <1 ms to 1 s, allowing a variation in send rate of 10 orders of magnitude. The TCP flow control attempts

to adaptively choose the window size using different algorithms, among them *Slow start*, *Congestion avoidance*, *Fast recovery*, and *Fast retransmission* (Erramilli et al. 2002).

3.5.2.1 TCP flows with congestion avoidance

TCP senders follow a self-limiting congestion avoidance algorithm that adjusts the size of a congestion window according to the inferred level of congestion in the network. The congestion window is adjusted by so-called additive increase multiplicative decrease (AIMD). In the absence of congestion when packets are acknowledged before their retransmission time-out, the congestion window is allowed to expand linearly. If a retransmission timer expires, TCP assumes that the packet was lost and infers that the network is entering congestion. The congestion window is collapsed by a factor of a half. Thus, TCP flows generally have the classic “saw tooth” shape. Accurate models of TCP dynamics are important because of TCP’s dominant effect on Internet performance and stability. Hence, many studies have attempted to model TCP dynamics (a significant synthesis is presented in Altman et al. 2005). Simple equations for the average throughput exist, but the problem of characterizing the time-varying flow dynamics is generally difficult due to interdependence on many parameters such as the TCP connection path length, round-trip time, and probability of packet loss, link bandwidths, buffer sizes, packet sizes, and number of interacting TCP connections.

3.5.2.2 TCP flows with active queue management

The TCP congestion avoidance algorithm has an unfortunate behavior where TCP flows tend to become synchronized. When a buffer overflows, it takes some time for TCP sources to detect a packet loss. In the meantime, the buffer continues to overflow, and packets will be discarded from multiple TCP flows. Thus, many TCP sources will detect packet loss and slow down at the same time. The synchronization phenomenon causes underutilization and large queues.

Active queue management schemes, mainly random early detection (RED) and its many variants, eliminate the synchronization phenomenon and thereby try to achieve better utilization and smaller queue lengths. Instead of waiting for the buffer to overflow, RED drops a packet randomly with the goal of losing packets from multiple TCP flows at different times. Thus, TCP flows are unsynchronized, and their aggregate rate is smoother than before. From a queuing theory viewpoint, smooth traffic achieves the best utilization and shortest queue.

In conclusion, the impacts of the TCP flow control mechanisms on the traffic are very strong. TCP-type feedback appears to have the effect of modifying the self-similarity traffic behavior, but it neither generates it nor eliminates it.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>