Fractal-Based Point Processes

2005

Steven Bradley Lowen
Harvard Medical School
McLean Hospital

Malvin Carl Teich
Boston University
Columbia University

WILEY

13

Computer Network Traffic

In the course of his studies of telephone traffic, **Agner Krarup Erlang (1878–1929)**, a Danish mathematician, conceived of a number of important point processes and established the fundamental framework for queueing theory.

The Swedish mathematician **Conny Palm (1907–1951)** advanced the approach set forth by Erlang by incorporating realistic features of telephone traffic, such as the clustering of calls and the superposition of traffic on multiple channels.

In this, the final chapter of the book, we show how the various approaches and models developed in previous chapters can be used to analyze **computer network traffic**, a process that is at the same time complex and rich in fractal behavior.[1] The mathematical study of computer network traffic is called **teletraffic theory**. This

[1] This chapter is not designed to provide a comprehensive introduction to computer network traffic in general, nor to its fractal characteristics in particular. For the latter, we refer the reader to the comprehensive tome compiled by Park & Willinger (2000), the excellent article by Abry, Baraniuk, Flandrin, Riedi & Veitch (2002), and the didactic book chapter authored by Willinger, Paxton, Riedi & Taqqu (2003).

theory encompasses various features of queueing theory, stochastic processes, control theory, optimization theory, and graph theory. In practice it proves useful for ensuring network stability and for the optimization of resource allocation. Teletraffic theory enables us to evaluate routing protocols and switch designs and offers a point of departure when planning network expansion. Agner Krarup Erlang is widely known as the “father of teletraffic theory.”

We begin in Sec. 13.1 with a brief review of early Poisson-based approaches to modeling **telephone network traffic**, as initially set forth by Erlang (1909), Engset (1915), and Palm (1937). In the course of this review, we provide an elementary introduction to queueing theory. In Sec. 13.2, we examine modern **computer communication networks**, which carry information in the form of packets,[2] and contrast these systems with telephone networks. We devote Sec. 13.3 to an examination of the fractal nature of computer network traffic. Various salient issues pertaining to modeling and simulation are set forth in Sec. 13.4. In Sec. 13.5 we consider a number of fractal-based point-processes that have served as models for computer network traffic.

Finally, in Sec. 13.6 we offer the reader a didactic step-by-step approach designed to assist in the identification of an unknown fractal-based point process. Using computer network traffic as an example, we demonstrate that the data sets we examine follow the form of a fractal-rate point process, and closely resemble the biological point process recorded at the striate cortex. Bearing in mind the tradeoff between model accuracy and parsimony, we conclude that two point-process models are good candidates for describing computer network traffic: a Neyman–Scott cluster process and a Bartlett–Lewis cascaded process. We examine the performance of these two models in some detail, and compare and contrast their predictions with two classic Ethernet-traffic data sets.

13.1 EARLY MODELS OF TELEPHONE NETWORK TRAFFIC

In the early years of telephone service, the subscribers in a town typically connected to a common exchange, staffed by an operator who routed all calls to their intended destinations. Routing calls within the town required only a simple connection at the exchange and rarely led to delay. However, call requests to numbers at other exchanges required the use of shared lines to those exchanges, and to additional lines and exchanges for very long-distance calls. When all lines to another exchange were busy, someone wishing to place a call through it would have to wait until one of the lines became free.

Service could, of course, be improved by installing an individual line for each customer, but the cost of doing so would be exorbitant. The intelligent design of

[2] Packets comprise small blocks of bits that travel together over a computer network, independently of other packets. Typically, the information in a file or data stream comprises a large number of packets.

any telephone network offers the engineer the following challenge: how to route calls among exchanges with a specified degree of reliability — and within a certain budget.

The design of an efficient system requires detailed knowledge of the offered load of call traffic. As part of a comprehensive examination of the applications of probability theory to telephone traffic in his native Denmark, Erlang carried out the first analyses of inter-exchange telephone traffic in 1909, 1917, and 1920.[3] He argued that the calls initiated by any one person form a negligible part of the aggregate call traffic at a large exchange. He also reasoned that different people initiate calls largely independently. Taken together, these heuristic arguments suggested that the homogeneous Poisson process (see Sec. 4.1) provides a suitable model for the aggregate traffic. And, indeed, this does turn out to be the case under many circumstances.

An extension of some of Erlang's results was provided by the Norwegian mathematician Engset, both in an unpublished manuscript completed in 1915 [Myskja (1998a) provides commentary on this manuscript] and in a paper published in 1918 [Jensen (1992) provides commentary on this paper].[4] In 1943, the Swedish mathematician Palm offered a number of significant generalizations of Erlang's results. He introduced such key features as slow rate modulations associated with daily, weekly, and yearly cycles; sudden increases in traffic following popular sporting events or major disasters; and the complexities of traffic that span multiple exchanges. The incorporation of these considerations played a crucial role in the design of efficient telephone networks.

13.1.1 Queueing theory

Queueing theory provides a suitable point of departure for studying simple telephone networks (Cohen, 1969; Cooper, 1972; Kleinrock, 1975; Asmussen, 2003). This mathematical formalism describes the utilization of a resource on which demands are made in a random fashion. The arrival times, and the magnitude of the resource requested per demand, may be random, and the resource itself may also vary in time. Demands that cannot be immediately met are queued (stored in a buffer) or declined.

For didactic purposes, we begin by considering the simple homogeneous Poisson-process model of call arrivals at a telephone exchange. Upon arrival, each call is queued. Resources can only be provided for storing a finite number of unprocessed call requests. As telephone lines come available, operators connect calls to their intended destinations in the order in which they arrived. The call durations follow an exponential distribution, corresponding to the interevent intervals of a homogeneous Poisson process. We consider the case of a single outgoing telephone line.

To model this call-activity sequence, we make use of the following construct:

[3] For a brief discussion of these papers, see Brockmeyer, Halstrøm & Jensen (1948, pp. 101–104). The 1917 paper is widely considered to be Erlang's most important.

[4] Engset (1915) highlighted the importance of the *truncated binomial distribution*, an extension of Erlang's (1917) *B formula*.

1. The **queue length** or **buffer occupancy** $Q(t)$ assumes integer values between a minimum of zero and a maximum of Q_m. The quantity Q_m is known as the **maximum queue length** or **buffer size**.

2. Calls arrive at times $t_{a,k}$ corresponding to a homogeneous Poisson process $N_a(t)$ with fixed, deterministic rate μ_a, where the label a denotes that it represents the **arrival process**, and k indexes the arrival times.

3. The service times are independent and identically distributed exponential random variables with mean duration $1/\mu_s$; the corresponding auxiliary homogeneous Poisson process $N_s(t)$ has a fixed, deterministic rate μ_s and corresponding event times $t_{s,k}$. The label s denotes that it represents the **service process**, and k again serves as an index.

4. When $Q(t) < Q_m$, $Q(t)$ increments by unity at each $t_{a,k}$.

5. When $Q(t) = Q_m$, the events of $N_a(t)$ correspond to dropped calls.

6. When $Q(t) > 0$, $Q(t)$ decrements by unity at each $t_{s,k}$.

In a handy notation developed by Kendall (1953), this model is called an M/M/1/Q_m queue (Kleinrock, 1975; Gross & Harris, 1998). The first symbol describes the **arrival process**, "M" for "Markov" in this case, indicating independent arrivals and therefore a homogeneous Poisson process. The exponentially distributed duration of each call corresponds to a homogeneous Poisson **service process**, so that "M" stands as the second symbol as well. The "1" that stands as the third symbol signifies the **number of servers** (outgoing lines). Finally, the last symbol "Q_m" characterizes the **maximum queue length**; by convention, the omission of this symbol signifies that $Q_m = \infty$. Other queueing models comprise different arrival or service processes, including those that are deterministic ("D") or general ("G"), and allow for an arbitrary number of servers.[5]

We now proceed to write a state equation for this model. Let $p_Q(n,t) \equiv p\,(n,t)$ represent the **queue-length distribution**, the probability that $Q(t) = n$. Except for the boundary cases $n = 0$ and $n = Q_m$, a constant rate of change μ_a associated with the arrival process $N_a(t)$ carries the queue-length distribution from $p\,(n,t)$ to $p\,(n+1,t)$, which concomitantly decreases $p\,(n,t)$. For this component we therefore have $dp\,(n,t)/dt = -\mu_a\,p\,(n,t)$. Similarly, an arrival when $Q(t) = n-1$ increases $p\,(n,t)$ via the term $+\mu_a\,p\,(n-1,t)$. The service process provides analogous contributions: $-\mu_s\,p\,(n,t) + \mu_s\,p\,(n+1,t)$. Recognizing that $p\,(n-1,t) = 0$ for $n = 0$ and $p\,(Q_m+1,t) = 0$ for $n = Q_m$ accommodates the boundary cases. Combining

[5] Poisson-arrival and exponential-service processes have traditionally provided a good description for the public switched telephone network. These assumptions can no longer be fully justified, however, because of the vast changes that have taken place in the voice telephone network in recent years, such as its increased use for internet connections and facsimile transmission (see, for example, Duffy, McIntosh, Rosenstein & Willinger, 1994).

all terms, including those for the boundary cases, then leads to a rate equation known as a **forward Kolmogorov equation**:

$$\frac{dp\,(n,t)}{dt} = \begin{cases} -\mu_a\, p\,(n,t) & \qquad\qquad\qquad\qquad + \mu_s\, p\,(n+1,t) & n = 0 \\ -\mu_s\, p\,(n,t) & \qquad + \mu_a\, p\,(n-1,t) & n = Q_m \\ -(\mu_a+\mu_s)\, p\,(n,t) + \mu_a\, p\,(n-1,t) + \mu_s\, p\,(n+1,t) & & 0 < n < Q_m. \end{cases} \tag{13.1}$$

Under steady-state conditions, the left-hand side of Eq. (13.1) is zero for all n. A bit of algebra then leads directly to the **geometric queue-length distribution** (Erlang, 1917; Palm, 1943),

$$p_Q(n,t) \to \frac{(1-\rho_\mu)\,\rho_\mu^n}{1-\rho_\mu^{Qm+1}}\,, \tag{13.2}$$

where the **service ratio** (also called **server utilization**) is defined as

$$\rho_\mu \equiv \frac{\mu_a}{\mu_s}\,. \tag{13.3}$$

For the special case of infinite buffer size, we recover the M/M/1/$\infty \equiv$ M/M/1 queue, in which case Eq. (13.2) reduces to

$$p_\infty(n,t) = (1-\rho_\mu)\,\rho_\mu^n. \tag{13.4}$$

For a service ratio $\rho_\mu = 0.9$, we display this geometric queue-length distribution as the dashed straight line in Fig. B.15 (semilogarithmic coordinates), and as the dotted curve in Fig. B.16 (doubly logarithmic coordinates).

Three measures turn out to be useful for assessing queueing-system performance: the **mean queue length** (or **mean number of waiting calls**), the **mean waiting time** spent in the buffer, and the **overflow probability**. We consider these measures in turn.

The mean number of waiting calls follows directly from the distribution provided in Eq. (13.2) (Palm, 1943):

$$\begin{aligned} \mathrm{E}[Q] &= \sum_{n=0}^{\infty} n p_Q(n) \\ &= \sum_{n=0}^{Qm} n \frac{(1-\rho_\mu)\,\rho_\mu^n}{1-\rho_\mu^{Qm+1}} \\ &= \frac{\rho_\mu - (Q_m + 1 - \rho_\mu Q_m)\,\rho_\mu^{Qm+1}}{(1-\rho_\mu)\left(1-\rho_\mu^{Qm+1}\right)}\,. \end{aligned} \tag{13.5}$$

Straightforward algebra yields the higher-order moments of this distribution as well.

An intuitive but nontrivial result, known as **Little's law** (Little, 1961), provides that the mean waiting time for a single server is simply the mean number of waiting calls multiplied by the mean service time:

$$\mathrm{E}[\tau_w] = \frac{\mathrm{E}[Q]}{\mu_s}\,. \tag{13.6}$$

Results for multiple servers are somewhat more complex, although still quite tractable, since call-traffic sharing occurs across lines; before any call encounters a delay, all lines must be occupied. We readily modify Eq. (13.6) to yield an approximate result for M servers:

$$\mathrm{E}[\tau_w] \approx \frac{\mathrm{E}[Q]}{M\mu_s}\,. \tag{13.7}$$

Nevertheless, we emphasize that Eqs. (13.1)–(13.5) change form for M servers. Transition rates among different queue occupancy probabilities $p\,(n,t)$ vary with n for $n < M$; not all M lines carry calls if fewer than M calls reside in the buffer. In particular, $\mathrm{E}[Q]$ no longer follows the form set forth in Eq. (13.5).

The third performance measure is the probability P_B that an arriving call fails to enter the buffer because it is full. This quantity is known as the **buffer overflow probability** (or **call-drop probability** or **blocking probability**). Setting $n = Q_m$ in Eq. (13.2) for the single server yields

$$\begin{aligned} P_B &= \lim_{t\to\infty} p_Q(Q_m,t) \\ &= \frac{(1-\rho_\mu)\,\rho_\mu^{Qm}}{1-\rho_\mu^{Qm+1}} && (13.8) \\ &= \frac{1-\rho_\mu}{\rho_\mu^{-Qm}-\rho_\mu}\,. && (13.9) \end{aligned}$$

The proportion of arrivals that finds the queue full equals the proportion of times that the queue is full. Said differently: *Poisson arrivals see time averages*, often captured by the acronym *PASTA* (Wolff, 1982). For large buffer sizes Q_m, the term ρ_μ^{-Qm} in the denominator of Eq. (13.9) dominates ρ_μ for $\rho_\mu < 1$, so that $\rho_\mu^{-Qm} - \rho_\mu \to \rho_\mu^{-Qm}$ [this approximation understates P_B by the factor $1/(1-\rho_\mu^{Qm-1}) \approx \rho_\mu^{Qm-1}$]. The overflow probability then reduces to

$$P_B \approx (1-\rho_\mu)\,\rho_\mu^{Qm} \sim\; \rho_\mu^{Qm}. \tag{13.10}$$

Equation (13.10) reveals that Poisson arrival and service processes give rise to an overflow probability that decreases with decreasing service ratio ρ_μ as a power-law function, and decreases with increasing maximum queue length Q_m as an exponential function.

Figures 13.1 and 13.2 display the behavior of the overflow probability P_B set forth in Eq. (13.10), as a function of the service ratio ρ_μ and of the maximum queue length Q_m, respectively. Relatively modest values of the maximum queue length yield quite small overflow probabilities. Erlang first presented these results, as well as exact results for M independent servers (telephone lines), in 1917.

For the M/M/1/Q_m queue, Eq. (13.2) shows that $p_Q(n) \sim \rho_\mu^n$ while Eq. (13.10) tells us that $P_B(Q_m) \sim \rho_\mu^{Qm}$. We conclude that for fixed ρ_μ, both the queue-length distribution and the overflow probability follow a geometric distribution. Indeed, for

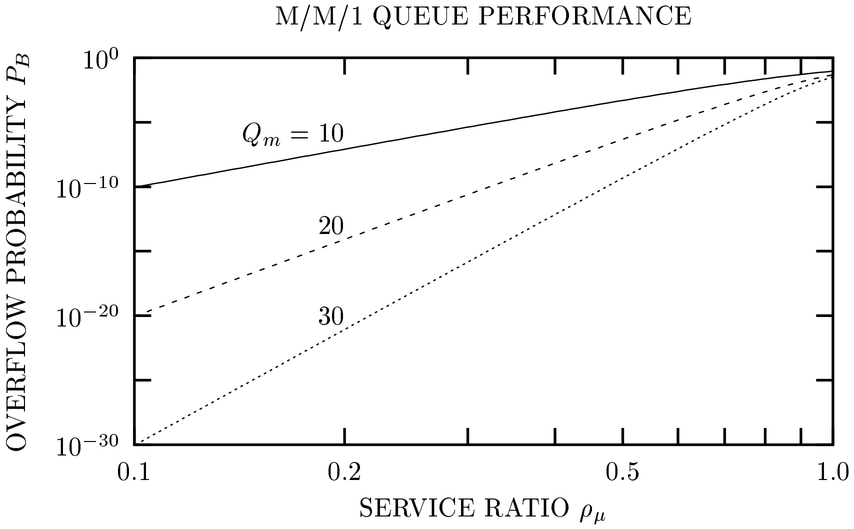

Fig. 13.1 Buffer overflow probability P_B as a function of the service ratio $\rho_\mu \equiv \mu_a/\mu_s$, for three values of the maximum queue length: $Q_m = 10$ (solid curve), 20 (dashed curve), and 30 (dotted curve). The roughly straight-line behavior on this doubly logarithmic plot represents the power-law relation between P_B and ρ_μ inherent in Eq. (13.10).

$Q_m \to \infty$, Eqs. (13.4) and (13.10) provide

$$\begin{aligned} p_\infty(Q_m) &= (1-\rho_\mu)\,\rho_\mu^{Qm} \\ P_B &\approx (1-\rho_\mu)\,\rho_\mu^{Qm}, \end{aligned} \tag{13.11}$$

respectively, where $p_\infty(n)$ represents the queue-length distribution for an M/M/1/$\infty \equiv$ M/M/1 queue. As discussed in Probs. 13.3 and 13.5, these equations demonstrate that the infinite-buffer queue-length distribution $p_\infty(n)$, evaluated at $n = Q_m$ where Q_m is the buffer size, provides an approximation for the overflow probability of the M/M/1/Q_m queue:

$$P_B \approx p_\infty(Q_m). \tag{13.12}$$

13.2 COMPUTER COMMUNICATION NETWORKS

Modern computer communication networks differ greatly from their voice-based precursors. Indeed, they are possibly the most complex of all systems contrived by humans. Data travel as small blocks of digital bits, in the form of packets, rather than as entities such as entire telephone conversations or files. No master scheduler directs the functioning of routers in the network; rather, each router passes packets

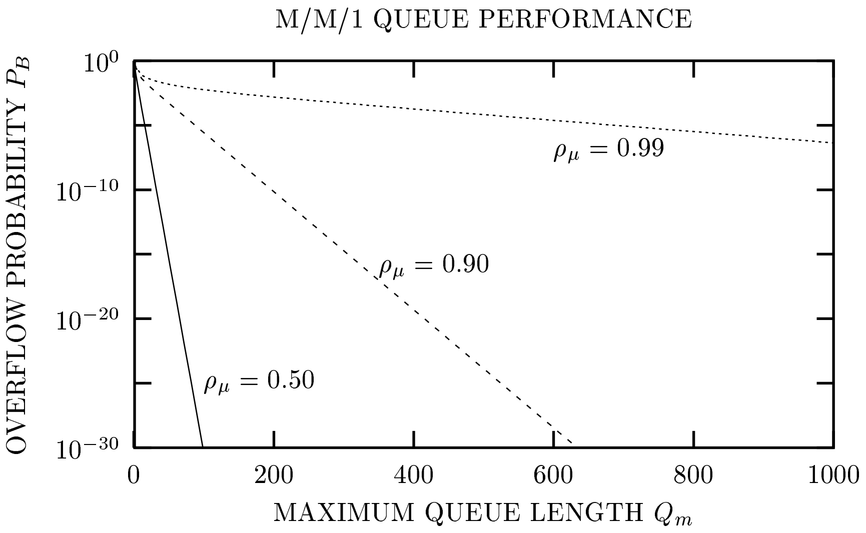

Fig. 13.2 Buffer overflow probability P_B as a function of the maximum queue length Q_m, for three values of the service ratio: $\rho_\mu \equiv \mu_a/\mu_s = 0.50$ (solid curve), 0.90 (dashed curve), and 0.99 (dotted curve). The roughly straight-line behavior on this semilogarithmic plot represents the exponential relation between P_B and Q_m inherent in Eq. (13.10).

on to other routers based largely on local activity and availability. The network itself dynamically allocates the routes over which the packets travel. As a consequence, packets flow smoothly around a blocked router, whereas a corresponding failure in a voice network might easily disable a large section of the network.

13.2.1 Scale-free networks

Both the Internet and the World Wide Web[6] behave as a scale-free networks [see Sec. 2.7.8 and Albert & Barabási (2002); Dorogovtsev & Mendes (2003); Pastor-Satorras & Vespignani (2004); Song, Havlin & Makse (2005)]. Such networks abound in the domain of computer communications — power-law distributions describe: (1) the number of edges emanating from a vertex in the Internet graph (Faloutsos, Faloutsos & Faloutsos, 1999; Aiello, Chung & Lu, 2001); (2) the number of exchanged emails per email address (Ebel, Mielsch & Bornholdt, 2002); (3) the number of web pages per website (Huberman & Adamic, 1999); and (4) the number of hyperlinks per web page in the virtual World Wide Web (Albert et al., 1999).

[6] The nodes of the Internet are the physical routers and computers while the edges are the connecting cables and wires. The nodes of the World Wide Web are web documents while the edges are the directed hyperlinks (URLs) that connect them.

Proper design of network topologies, and avoiding the deleterious effects of coordinated attacks against network hubs, require that we understand and accommodate the scaling nature of the network.

13.2.2 Static representation

Even a static representation of the Internet proves difficult to analyze. Figure 13.3 shows one representation of the major ISP (Internet Service Provider) nodes of the Internet, indicated as small squares. The angular position around the circle indicates the geographical longitude of the node while the distance from the center to each node varies inversely with the traffic carried by that node. The Internet comprises more than 100 000 separate networks with more than 100 million hosts. There are millions of routers, billions of web locations, and tens of billions of catalogued documents resident on the World Wide Web.

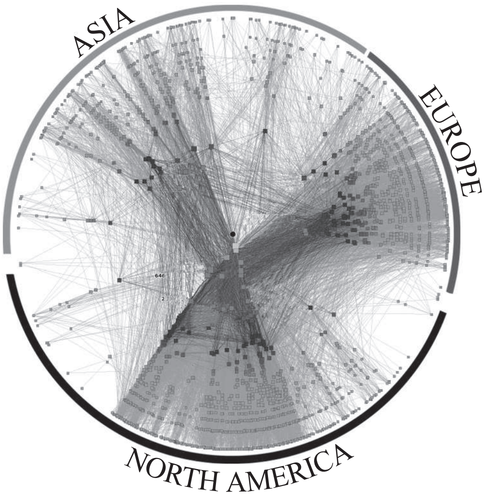

Fig. 13.3 Snapshot of ISPs (Internet Service Providers) constructed from data collected during the period 21 April 2003 through 8 May 2003. The angular position around the circle represents the geographical longitude of the ISP node (represented by a small square) while the distance from the center to each node varies inversely with the traffic carried by that node. The graph reflects more than 1 million IP (Internet Protocol) addresses and more than 2 million IP links that are, roughly speaking, aggregated into a topology of 11 000 ISPs. Adapted from http://www.caida.org/analysis/topology/as_core_network/, which provides details of this representation.

13.2.3 Vertical layers

In conjunction with the horizontal complexity of the Internet described above, information in computer communication networks is transmitted in a vertically rich manner, usually using a five-layer TCP/IP (Transmission Control Protocol/Internet Protocol) suite. Each layer relies on the layer below it for executing more primitive functions, while providing services to the layer above it. The highest layer corresponds to applications such as HTTP, whereas the lowest layer handles the physical transfer of bits over the medium.

We thus consider teletraffic in terms of five layers, each with its own set of tasks and protocols (conventions and rules):

- *Application Layer*: Execution of individual applications such as HTTP (hypertext transfer protocol), FTP (file-transfer protocol), TELNET (telephone network remote connection), or SSH (secure shell).

- *Transport Layer*: Delivery of events within those applications, such as individual file transfers within an HTTP session (TCP is the transport protocol for TCP/IP).

- *Internetwork Layer*: Transmission through the Internet of blocks of packets within those file transfers (IP is the internetwork protocol for TCP/IP).

- *Link Layer*: Transmission of individual packets within those blocks of packets on individual links.

- *Physical Layer*: Transmission of individual bits within those individual packets on a particular link.

In general, different vertical layers exhibit different statistics. Users initiating HTTP sessions, for example, might well follow a homogeneous Poisson process, at least over time scales of an hour or less (Feldmann, Gilbert & Willinger, 1998). On the other hand, the initiation times for individual HTTP commands, such as requests for documents or images, would likely follow a different statistical pattern. For example, individual packet arrivals might be characterized by a fractal-based point process as a result of power-law-distributed file sizes (see Sec. 13.3.5).

Figure 13.4 displays a highly schematized picture of information transmission on such a multi-layered structure, in the form of a cascaded point process. The primary point process $dN_1(t)$ in a) might represent the arrivals of HTTP file-transfer requests to a server. Each secondary point process $dN_{2,k}(t)$ in b) would then describe the resulting packet transfers for the corresponding files measured at a nearby downstream node, with the number of packets or temporal duration of each secondary process corresponding to the extent of the associated file-transfer flow. The total packet traffic process $dN_3(t)$ displayed in c) might then comprise the superposition of all packet arrival times at that nearby node.

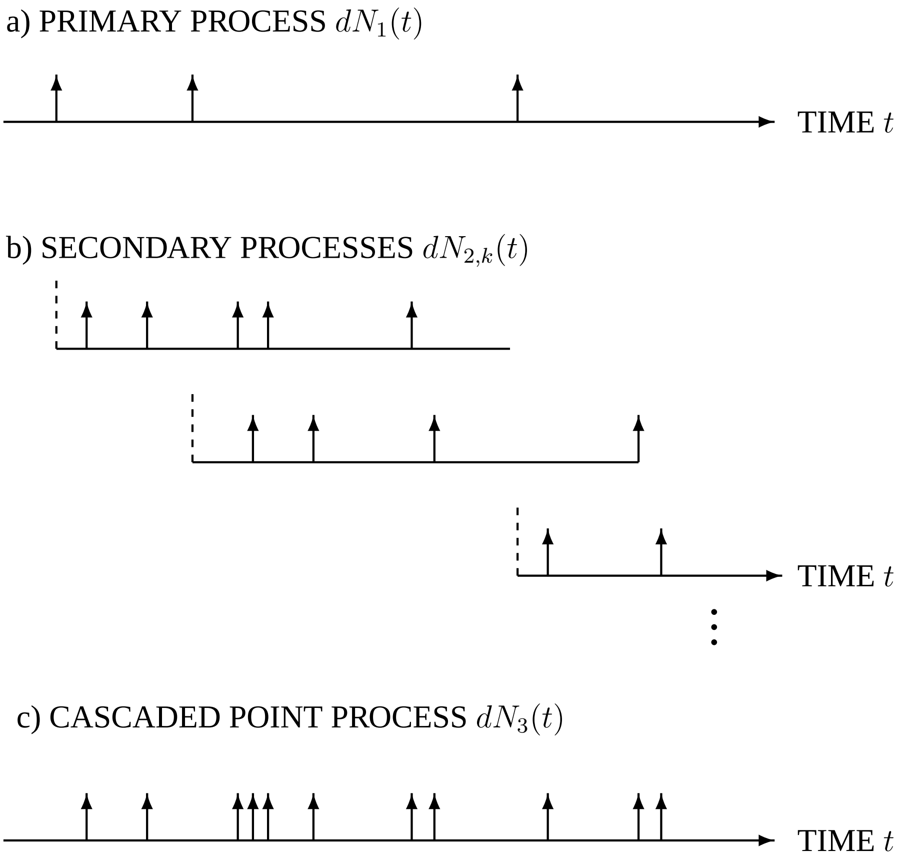
a) PRIMARY PROCESS $dN_1(t)$
TIME t
b) SECONDARY PROCESSES $dN_{2,k}(t)$
TIME t
c) CASCADED POINT PROCESS $dN_3(t)$
TIME t

Fig. 13.4 Partial schematic for computer network traffic based on a cascaded point process (see Fig. 4.2 and Sec. 4.5). Each event of a primary point process $dN_1(t)$ (displayed in a) initiates a secondary point process $dN_{2,k}(t)$ that terminates after a random number of events or a random duration (displayed in b). All secondary points, taken together as indistinguishable events, form the cascaded-point-process output $dN_3(t)$ (displayed in c). Special cases of cascaded point processes include the fractal Bartlett–Lewis process (Sec. 10.6.4) and the fractal Neyman–Scott cluster process (Chapter 10).

13.3 FRACTAL BEHAVIOR

Designers of the first computer communication networks attempted to emulate the approach used for voice networks, borrowing equations and even terminology from telephony. Telephone lines became links in computer networks, exchanges became servers, and calls became, variously, data streams, files, or packets. However, early results proved disappointing. Small increases in buffer size did not dramatically reduce the overflow probability for computer communication networks, as would be expected on the basis of Fig. 13.2.

Examining the packet streams revealed that computer network traffic arrived in unpredictable bursts of activity over many time scales. To accommodate this behavior, researchers proceeded to formulate increasingly complex Markov models, but with limited success. These models relied on the implicit assumption that fluctuations in the offered load resemble those of a homogeneous Poisson process for time scales beyond a manageable cutoff time. But no such cutoff appeared to exist. Moreover, as described in Sec. 13.2, the topology of the Internet and the dynamics of the World Wide Web are constantly in flux. Unusual features such as these have far-reaching implications for network engineering (Taubes, 1998).

13.3.1 Early evidence

In 1993, Leland and colleagues presented a seminal paper, followed a year later by an extended version (Leland et al., 1994), in which they demonstrated that the rate of Ethernet traffic varied as a fractal process with long-range dependence; these authors further suggested that Poisson behavior does not obtain at any useful time scale. Many subsequent measurements of computer communication traffic have vetted this early finding (see, for example, Willinger et al., 2003, and references therein), demonstrating that fractal behavior over a large range of time scales is present in many different kinds of traffic: Ethernet local-area-network (LAN) traffic (Leland et al., 1994); wide-area-network (WAN) traffic (Paxson & Floyd, 1995), variable-bit-rate (VBR) video traffic (Beran, Sherman, Taqqu & Willinger, 1995); and World Wide Web (WWW) traffic (Crovella & Bestavros, 1997).

Soon after the first of these results appeared, traffic models based on fractional Brownian motion (Norros, 1995) revealed that classical Poisson-based techniques provided seriously flawed predictions for such systems. The queue-length distributions and overflow probabilities turned out to decrease far more slowly with buffer size than expected on the basis of the exponential functions displayed in Fig. 13.2. Markov models can generate highly variable traffic loads ("burstiness") over short time scales but the variability always diminishes as the time scale increases. Traffic with fractal characteristics, on the other hand, exhibits significant fluctuations at all time scales, with concomitant high-rate periods of all durations.

13.3.2 Second-order statistics

To illustrate the fractal nature of computer network traffic, we analyze the classic Ethernet local-area-network (LAN) data set BC-pOct89. The data comprise the arrival times and durations of the first 1 million packets recorded on the main Ethernet cable at the Bellcore (BC) Morristown Research and Engineering Facility over a period of about 29 minutes beginning at 11:00 AM on 5 October 1989 (Leland & Wilson, 1989, 1991). We initially examine the rate spectrum and the normalized Haar-wavelet variance, measures that prove to be highly useful for parameter estimation, as discussed in Sec. 12.4. We subsequently examine a whole raft of statistical measures for these data (see Sec. 13.6.1).

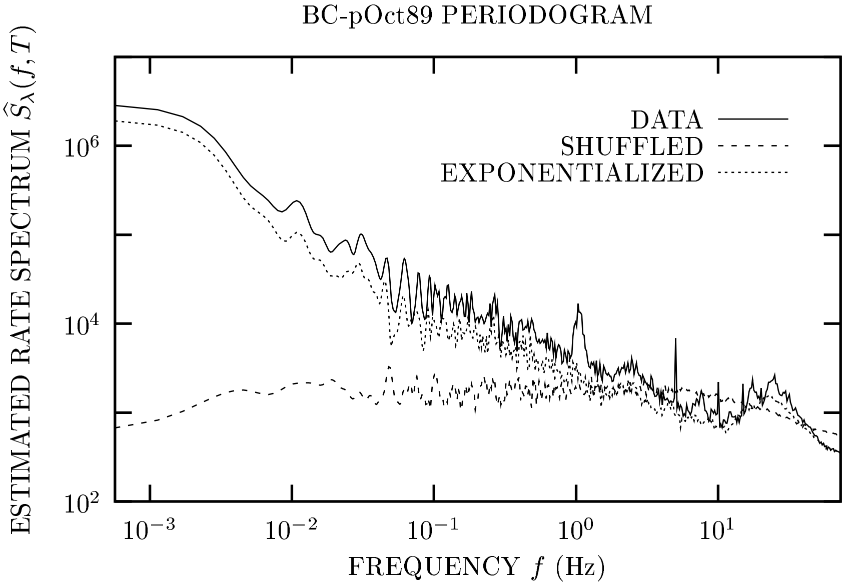

Fig. 13.5 Estimated rate spectrum (periodogram) $\widehat{S}_\lambda(f,T)$ vs. frequency f for the BC-pOct89 data set (solid curve), as well as for its exponentialized (dotted curve) and shuffled (dashed curve) surrogates. We smoothed the spectral estimate using the procedure reported in Footnote 7 on p. 117. The more-or-less straight-line decrease of the solid curve suggests that BC-pOct89 has a fractal rate. Since exponentialization leaves the fractal behavior only slightly changed, while shuffling destroys it, we conclude that the fractal behavior derives from the ordering of the intervals rather than from their distribution. Periodic components are in evidence at a number of frequencies.

As shown in Fig. 13.5, the estimated rate spectrum (solid curve) decreases in a power-law fashion over a broad range of frequencies f, confirming the presence of $1/f$-type noise and fractal behavior.[7] The periodogram of the exponentialized intervals (dotted curve; see Sec. 11.4.2) resembles the periodogram of the original data. In contrast, a shuffled version of the data (dashed curve; see Sec. 11.5) yields a periodogram that is devoid of power-law behavior. These results collectively indicate that the relative ordering of the intervals, rather than their distribution, is responsible for the fractal character of the data. Using the same reasoning we conclude that the broad spectral feature near $f = 30$ Hz derives largely from the interval ordering.

[7] The ordinate and abscissa are unnormalized in the BC-pOct89 periodogram displayed in Fig. 13.5. These same data appear in Fig. 5.1 with both the ordinate and abscissa normalized, and in Fig. 13.7i) with only the ordinate normalized.

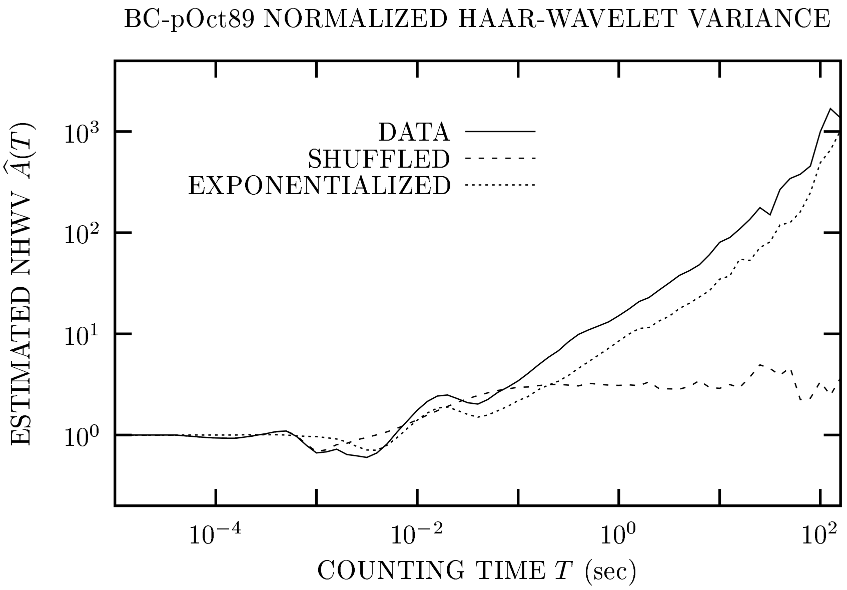

Fig. 13.6 Estimated normalized Haar-wavelet variance $\widehat{A}(T)$ vs. counting time T for the BC-pOct89 data set (solid curve). As with $\widehat{S}_\lambda(f, T)$, shown in Fig. 13.5, the more-or-less straight-line behavior suggests that BC-pOct89 has a fractal rate. Again, the surrogate data indicate that the interval ordering, rather than the interval distribution, is responsible for the fractal character of the data.

The estimated normalized Haar-wavelet variance (see Sec. 3.4.3) displayed in Fig. 13.6 (solid curve) also follows a power-law form over a broad range of counting times T, thereby confirming the conclusions drawn from the periodogram in Fig. 13.5. Computing this statistic for the two surrogate data sets also confirms that the relative ordering of the intervals, rather than their distribution, generates the fractal behavior. The broad bump in $\widehat{A}(T)$ near $T = 0.02$ sec corresponds to the spectral feature near $f = 30$ Hz in Fig. 13.5. The refractory behavior evident near $T = 0.005$ sec corresponds to frequencies that lie above the upper limit of the periodogram in Fig. 13.5.

13.3.3 Queueing-theory analysis

As a consequence of its fractal character, the second-order statistics of teletraffic do not follow Markov predictions, as shown in Sec. 13.3.2. Nor does the queueing behavior, as we now proceed to demonstrate.

For negligibly small buffers, fractal behavior has little impact on queueing performance since short-term (nonfractal) fluctuations overwhelm the buffer resources (Grossglauser & Bolot, 1996; Ryu & Elwalid, 1996). At the opposite extreme, exceptionally large buffers rarely overflow. For intermediate buffer sizes, however, the fractal nature of the traffic adversely affects queueing performance.

The queue-length distribution provides a useful window on network performance for this commonly encountered situation. Queue-length distributions resulting from fractal arrivals, or heavy-tailed service times, decay slowly with queue length in comparison with Markov predictions, often as power-law or Weibull functions (see, for example, Cohen, 1969, 1973; Norros, 1994; Brichet, Roberts, Simonian & Veitch, 1996; Roughan, Veitch & Rumsewicz, 1998; Asmussen, 2003). This has important implications for computer network traffic and for the design of computer communication networks (Erramilli, Narayan & Willinger, 1996).

A queue-length histogram that follows a decaying power-law form appears as the solid curve in Fig. B.16 (the solution to Prob. 13.6). This simulated result derives from the FGPDP/M/1 queue ($\rho_\mu = 0.9$), for which a fractal-Gaussian-process-driven doubly stochastic Poisson process (FGPDP) describes the arrivals, and the service times are exponential. We focus on this particular queue because of the ubiquity and importance of this arrival process (see Secs. 6.3.3, 8.4, 10.6.1, 13.5.3, and Chapter 12). Moreover, these results closely approximate those for the rectangular fractal-shot-noise-driven Poisson process (RFSNDP) (see Fig. B.19, the solution to Prob. 13.8), a plausible model for computer network traffic as discussed in Secs. 13.5.5 and 13.6. This latter queue-length histogram is also equivalent to a queue comprising Poisson flow arrivals and heavy-tailed service times, as discussed in Sec. 13.4.3.

These power-law queue-length histograms differ sharply from their M/M/1 geometric cousins (dotted curves in Figs. B.16 and B.19), which emerge when a homogeneous Poisson process describes arrivals at the queue, and the service times are exponential. The arrival process evidently imparts its fractal character to the resulting FGPDP/M/1 and RFSNDP/M/1 queue-length histograms, yielding power-law forms for these relations (straight lines on doubly logarithmic plots, as shown in Figs. B.16 and B.19).

The net result is a far larger range of possible queue lengths for the fractal queues. No characteristic size exists beyond which overflow probabilities decrease dramatically with increasing buffer size. Ensuring that fractal traffic reaches its destination, rather than encountering buffer overflows, thus demands far larger buffers than those needed for traffic based on Markov processes. Furthermore, the buffer-size requirements depend critically on the value of the fractal exponent that characterizes the offered traffic. Finally, we cannot fully describe the network by the service ratio ρ_μ, since this quantity effectively fluctuates.

In addition to the first-order "quality-of-service" measures of network performance considered above, second-order queueing statistics also prove important for characterizing fractal computer network traffic (Park, 2000). These include the standard deviations of the queue waiting time ("jitter") and of the message-loss probability, which, in many cases, can greatly exceed their mean values by virtue of the large fluctuations imparted by the fractal rate.

Finally, we note that the multilayered structure of commonly used protocols adds complexity to quality-of-service specifications (Park, 2000). Each layer has its own communication structure, and therefore a different set of statistics to specify. For example, specifications for the application layer might include file transfer rates for FTP or latency for an SSH session, but for the link layer they may involve packet-drop probabilities.

13.3.4 Predictability

The discussion provided in Sec. 13.3.3 shows that buffer overflows in computer communication networks stem from the persistence of fractal-rate fluctuations; a rate above the mean will likely remain so for some time. However, this same persistence leads to predictability in the traffic flow and can therefore be used to facilitate the allocation of resources to meet future needs. This approach can be useful in dynamically configuring network topologies; reallocation times can easily exceed buffer overflow times yet still lie well below the duration of long-term fluctuations.

Indeed, researchers have reported the feasibility of using predictive congestion control for fractal computer network traffic (Tuan & Park, 2000). However, different models yield vastly different values for the predictability,[8] and different estimates of fractal exponents also lead to varying results. Careful model choice and parameter estimation prove crucial in taking advantage of the predictability of such traffic.

13.3.5 Origins

As mentioned in Sec. 2.7.1, fractal behavior in computer network traffic is often ascribed to the power-law-distributed nature of file sizes (Park et al., 1996; Crovella & Bestavros, 1997). Imagine transferring a collection of such files over a computer communication network. Transfer via TCP (transmission control protocol) or UDP (user datagram protocol), in conjunction with flow and reliability control mediation, yields traffic with fractal properties (Park et al., 1996). In many cases, however, the fractal nature of computer network traffic appears to depend on all of the features involved. Eliminating flow and reliability controls by using UDP alone, for example, gives rise to output traffic that lacks much of the fractal structure of the input traffic (Park, Kim & Crovella, 2000). Moreover, the exponents of the file-size distributions do not always linearly predict the fractal exponents of the ensuing computer network traffic (Park et al., 2000). Evidently, the flow-control process can impart considerable complexity to the network traffic, beyond that of a simple fractal model.

Furthermore, large numbers of data-transfer processes occur concurrently within the same network, dividing network resources among them. Higher throughput for one application means less throughput for others. Features of one data stream, including the fractal exponents of its flow, may therefore ultimately derive from features of other streams. This interconnection becomes most important over times smaller than the round-trip time of the network under consideration. The melding of various fractal components may well lead to the multifractal characteristics seen over these shorter time scales (Willinger et al., 2003).

[8] We consider the predictability of fractional Brownian motion in Prob. 6.3.

Requests for power-law distributed traffic, mediated via lower-level transfer protocols, can lead to fractal behavior in communication networks. However, higher-level elements of the traffic stream can generate fractal fluctuations more directly. For example, VBR video traffic appears to exhibit fractal characteristics in a unified manner, and to lack meaningful discrete elements that themselves have power-law statistics (Garrett & Willinger, 1994). This fractal behavior may in turn derive from fractal characteristics in the input video stream.[9] Simple explanations do not always apply for such a richly complex system as the Internet.

13.3.6 Cutoffs

As discussed in Secs. 2.3.1 and 12.2.1, cutoffs play an important role in modeling fractal point processes; computer network traffic is no exception. The lion's share of the research in this connection concerns the long-time limit, and much of the mathematical framework that has been developed depends on the fractal behavior extending to infinite times. However, there are two reasons why fractal characteristics cannot extend to arbitrarily large times for real network traffic: (1) daily, weekly, and yearly rhythms exist, interfering with pure fractal behavior; and (2) all data sets truly have finite duration. Moreover, the absence of cutoffs leads to unwieldy mathematics. None of the results that we have set forth in this or earlier chapters depend on fractal activity extending to infinite times. Following the approaches specified in Secs. 2.3.1 and 12.2.1, we continue this tradition and employ finite cutoffs. Nevertheless, compelling mathematical reasons exist in some cases suggesting that outer cutoffs should be eliminated (Mandelbrot, 1997), especially in computer network traffic (Willinger, Alderson & Li, 2004).

13.3.7 Rate-process and point-process descriptions

As illustrated by the canonical data set labeled BC-pOct89, fractal fluctuations often form the salient characteristic of computer network traffic while effects at shorter time scales are less significant. Consider, for example, the normalized Haar-wavelet variance presented in Fig. 13.6 for these data. The plot manifests little evidence of fractal activity for counting times below about 10 msec, yet fully 98% of the interevent intervals lie below this value. Furthermore, some (although certainly not all) computer communication data sets have mean interevent intervals that lie far below the timestamp resolution. For example, the World Cup 1998 access log (Arlitt & Jin, 1998) comprises some 1 352 804 107 requests collected over 88 days from 30 April through 26 July 1998, at a resolution of one second. This translates to an average rate of 17.8 requests per second over the entire log; daily averages exceeding 81 requests per second (30 June); and correspondingly higher local rates over shorter time scales. Indeed, reconstructing the underlying point process for the World Cup log proves impossible.

[9] Natural scenes exhibit *spatial* fractal behavior in their own right, as discussed in Sec. 2.8.5.

We conclude that rate-based models, known as **fluid-flow models** in the context of computer network traffic, are highly useful. Many key results in fractal network traffic make use of this formulation, including early results relating to fractional Brownian motion (Norros, 1995).

We do not suggest, however, that rate-based models are superior to point-process models. As will become apparent in Sec. 13.6, data sets BC-pOct89 and BC-pAug89, which comprise experimental point processes, are readily analyzed as such.

13.3.8 Multifractal features

Computer network traffic comprises a multitude of events over a large range of time scales. In the BC-pOct89 data set analyzed in Figs. 13.5 and 13.6, for example, fully 5% of the intervals lie below 104 μsec; since the total duration of this data set exceeds 29 minutes, it effectively spans more than seven orders of magnitude in time.

With such a wide range of scales, detecting two or more scaling exponents becomes feasible.[10] Indeed, researchers have detected multifractal properties in wide-area network traffic over short time scales (Riedi & Lévy Véhel, 1997; Lévy Véhel & Riedi, 1997; Mannersalo & Norros, 1997). In particular, Feldmann et al. (1998) demonstrated that a conservative cascade model (Mandelbrot, 1974) could be used to characterize such traffic; however, they argued that multifractal properties exist only for (short) time scales that lie below the typical packet round-trip time (Riedi & Willinger, 2000). Taken together, this suggests that wide-area network traffic flow behaves as a multiplicative process over these short times, but becomes additive over longer time scales (Riedi & Willinger, 2000). Using this flow as a rate leads to the multiplicative-rate point process discussed in Sec. 5.5.1 (Schmitt et al., 1998). Interestingly, and in contrast, local-area network traffic appears to exhibit only a single fractal exponent and is therefore monofractal (Taqqu, Teverovsky & Willinger, 1997).

Other multifractal formalisms include processes with fractal exponents that vary with time, or across different realizations of the random process (Abry et al., 2000). We can readily construct a multifractal version of the fractal-shot-noise-driven Poisson process set forth in Chapter 10. Generalizing the fractal impulse response function defined in Eq. (9.2) provides

$$h(K,t) \equiv \begin{cases} f(K)\, t^{-\beta(K)} & A < t < B \\ 0 & \text{otherwise,} \end{cases} \tag{13.13}$$

where $\beta(K)$ and $f(K)$ are functions of the random variable K that describe the fractal exponent and its relative strength, respectively, for that particular impulse response function. In spirit, this approach resembles that used to generate fractal behavior from

[10] This stands in contrast to essentially all of the other fractal point-process data examined in earlier chapters, which had far shorter durations. In general, we found that characterizing one fractal exponent involved sufficient difficulty that other exponents, were they present, remained essentially undetectable.

a continuous superposition of Lorentzian spectra representing relaxation processes with time constants distributed over a range of values (see Sec. 2.7.9).

To analyze all putative multifractal processes, it proves helpful to access an extended range of statistical measures, such as the higher-order moments of wavelet transforms (Abry et al., 2000), $\mathrm{E}\big[\,|C_{\psi,N}(T,\cdot)|^q\big]$, and generalized dimensions D_q for many values of q (see Sec. 3.5.4).

13.4 MODELING AND SIMULATION

We now examine a number of salient issues pertaining to the mechanics of modeling and simulation. In Sec. 13.4.1 we contrast the use of models for analysis and synthesis; in Sec. 13.4.2 we provide a discussion of simulation methods and model complexity; and in Sec. 13.4.3 we highlight the fact that equivalent models can appear in different guises.

13.4.1 Analysis and synthesis

Models prove useful for both the analysis and synthesis of computer network traffic. The use of models for analysis helps us visualize the effects of various parameters on traffic flow and, in particular, assists us in identifying sources of fractal characteristics in the traffic stream.

The synthesis of computer network traffic, on the other hand, provides synthetic data that is invaluable for studying and testing yet-to-be-developed computer-communication-network protocols and topologies. It is hard to overstate the value of synthesis because of the prodigious volume of traffic data required to meaningfully evaluate a new network design. For example, establishing a message-loss probability of 10^{-9} with reasonable precision requires far more than 10^9 packets for a memoryless data stream, using a brute-force approach. Although good approximate methods requiring less data do exist, simulation sizes remain considerable. The use of realistic (fractal) traffic data, with its concomitant clusters of high- and low-activity periods, further increases data requirements. Evaluating such a network over a range of parameters easily involves terabytes of data. In the face of such vast requirements, the synthesis of data from fractal models is often the only viable alternative for evaluating performance, particularly for novel networks that have not yet been implemented.

13.4.2 Simulation approaches and model complexity

What methods prove best for simulating computer network traffic? One of the first issues that arises is whether to impose a feedback loop from the network under evaluation to the simulated incoming traffic. In other words, should the simulated traffic source change what it offers on the basis of network parameters such as queue length or number of dropped messages? In many real-world applications, such as video and audio streaming (see Sec. 13.3.5), the offered load does not depend on the state of the network. In other applications, however, network performance does affect the input traffic; users encountering excessive delays often terminate their connections and wait until the network becomes less busy.

We do not address this issue explicitly for our network traffic models since none of the processes we have considered to this point has provision for such feedback. But we note that implementing feedback of this kind is not difficult since every model has parameters that directly control the output rate. Indeed, as discussed in Sec. 13.3.5, evidence exists that some of the fractal behavior inherent in computer network traffic may derive from the flow-control process (Park et al., 2000).

After determining the broad class of simulation, the issue becomes the level of detail to incorporate in the model. A tradeoff always exists between reality and simplicity; the ideal model captures all salient features of a data set on the one hand, and yet derives simply from a few underlying principles on the other hand. Useful models must strike a balance on the continuum between these two ideals. In the context of computer network traffic, the vast quantities of data shift the optimal model strongly toward parsimony. Complex models do indeed capture more features of the data, but they also tend to assume an *ad hoc* nature and require significant efforts to program. Indeed, simulation execution times can grow out of bounds so rapidly that models including even modest complexity can become useless.

13.4.3 Equivalent models in different guises

Identical network traffic can sometimes appear in different guises, as we briefly mentioned in Sec. 13.3.3. We saw such a duality in the context of general point processes in Sec. 4.5: under certain conditions, cascaded and doubly stochastic representations offer two different formalisms for the same underlying point process.

Consider, for example, a collection of data flows that follow a Poisson arrival process and exhibit heavy-tailed durations. The overall flow is then fractal shot noise with a rectangular impulse response function. We expect similar service times for all packets since they are restricted to a maximum size and most are at or near that maximum. The constituent packets consequently behave as a fractal-shot-noise-driven Poisson process (or a closely related integrate-and-reset version thereof). This leads us to recognize that the two processes are therefore different descriptions of precisely the same traffic. We conclude that the rectangular fractal-shot-noise-driven Poisson process (fractal Neyman–Scott process) is equivalent to a queue comprising Poisson flow arrivals and heavy-tailed service times. The connection is most valuable since this latter queue has been studied extensively in the literature. Moreover, in some cases one formulation may prove conceptually simpler than another, or it may offer faster simulations.

13.5 MODELS

A brief overview of the structure of computer communication networks appeared in Sec. 13.2, and we considered the fractal character of the resident traffic flow in Sec. 13.3. We discussed a number of salient issues pertaining to modeling and simulation in Sec. 13.4.

With this background, we are now in a position to consider several of the models presented in earlier chapters in the context of computer network traffic. These models, which offer different balances between reality and simplicity, as discussed in Sec. 13.4.2, prove useful in elucidating the behavior of computer network traffic. The mere fact that we consider *several* models, however, highlights the difficulties associated with identifying a unique fractal-based point process for a given collection of data. New models continue to be set forth (see, for example, Field, Harder & Harrison, 2004a,b).

13.5.1 Fractal renewal point process

A fractal renewal point process (Chapter 7) serves as a suitable model for the activity associated with a single network-traffic application. The superposition of a number of these processes (see Sec. 11.6.2) then represents the aggregate traffic from a collection of such applications, and thus provides a useful model for computer network traffic (Ryu & Lowen, 1996, 1998). The power-law decaying interevent-interval distribution imparts fractal fluctuations to the simulated teletraffic. A useful generalization of this approach, which takes the form of a marked-point-process model, accommodates messages of different (often power-law distributed) sizes (Levy & Taqqu, 2000).

13.5.2 Alternating fractal renewal process

The alternating fractal renewal process (Chapter 8) also leads to useful models for computer network traffic (Ryu & Lowen, 1996, 1998). Rather than each message forming a point event, as considered in Sec. 13.5.1, periods during which $X(t) = 1$ correspond to a message (such as a TCP connection) whereas periods during which $X(t) = 0$ correspond to inter-message quiet. The sum of a number of such alternating fractal renewal processes then represents messages independently generated by several applications. This sum, which is fractal binomial noise (see Sec. 8.3.1), serves as the rate process for packet generation.

Both Poisson and integrate-and-reset packet-generation mechanisms prove useful. Specific results are available for the queue-length distribution (Boxma, 1996) and the buffer overflow probability (Ryu & Lowen, 1997, 1998) for these point processes, as well as for their rate-based approximations (Heath, Resnick & Samorodnitsky, 1998; Jelenković & Lazar, 1999). Both approaches illustrate the sometimes paradoxical effects of fractal behavior: for $1 < \gamma < 2$, the dwell times for $X(t)$ have mean values that are finite, yet the average quantity of data residing in a buffer, waiting to be transmitted, becomes infinite (Boxma, 1996). Also, in some cases buffer overflow probabilities turn out not to depend significantly on the rate at which messages leave the buffer (Heath et al., 1998). The alternating fractal renewal process may be particularly suitable for modeling HTTP activity; file sizes for this application are generally power-law distributed (Feldmann, Gilbert, Willinger & Kurtz, 1998) and users often alternate between web-page downloading $[X(t) = 1]$ and viewing $[X(t) = 0]$.

The extended alternating fractal renewal process (Yang & Petropulu, 2001; Yu, Petropulu & Sethu, 2005), in which the packet-generation rate alternates between zero and a random value, with all such values independent of each other, adds flexibility to the rate process for modeling the burstiness of computer network traffic. A related model that lacks the explicit final Poisson process has also been extensively investigated (Mandelbrot, 1969; Taqqu & Levy, 1986; Levy & Taqqu, 2000).

13.5.3 Fractal-Gaussian-process-driven Poisson process

The fractal-Gaussian-process-driven Poisson process is ubiquitous because many fractal-based point processes, as well as superpositions thereof, converge to it (see Secs. 6.3.3, 8.4, 10.6.1, 11.6.1, 13.3.3, and Chapter 12). Kurtz (1996) showed that similar behavior is observed in computer network traffic for the fractal-shot-noise-driven Poisson processes considered in Sec. 13.5.5.

If a number of traffic sources aggregate to produce an overall traffic stream, flow control will not significantly affect any one source by itself. Over long time scales, then, the fractal Gaussian process (Sec. 6.3.3) should provide a useful model for the rate of the resulting system. A number of queueing results based on such processes exist in the literature (Norros, 1995; Lévy Véhel & Riedi, 1997).

13.5.4 Fractal Bartlett–Lewis point process

The vertical-layer structure discussed in Sec. 13.2.3 suggests that cascaded-point-process models (see Fig. 13.4) may be useful for characterizing computer network traffic. We consider two such models, in turn: the fractal Bartlett–Lewis point process and the fractal Neyman–Scott point process.

The Bartlett–Lewis point process introduced in Sec. 4.5 makes use of a primary homogeneous Poisson process; the secondary processes comprise segments of renewal processes with independent intervals drawn from identical distributions. The homogeneous Poisson process does indeed provide a good description for session arrivals in some forms of computer network traffic, at least over time scales of an hour or less; examples include FTP and TELNET (Paxson & Floyd, 1995), as well as HTTP (Feldmann et al., 1998). In the fractal version of the Bartlett–Lewis model introduced by Grüneis and colleagues (see, for example, Grüneis, 1984; Grüneis & Baiter, 1986; Grüneis, 2001), which was discussed in Sec. 10.6.4, the number of events M_k in each secondary process follows a power-law form. The power-law-distributed nature of file sizes (Park et al., 1996; Crovella & Bestavros, 1997) accords with this model.

Hohn, Veitch & Abry (2003) considered the fractal Bartlett–Lewis model in the context of computer network traffic. To match the measured interevent-interval histogram, they drew renewal-process segments from identical gamma distributions[11] of order $\mathsf{m} = 0.60$ (see Prob. 4.7). They selected the number of events M_k in each secondary process to follow a power-law form with no upper scaling cutoff, and chose the exponent such that M_k had finite mean but infinite variance. The model of Hohn et al. (2003) thus has five parameters: the primary-process mean interevent interval $\mathrm{E}[\tau_1]$, the secondary-segment mean interevent interval $\mathrm{E}[\tau_2]$, the order of the gamma distribution m for this interevent interval, the mean number of events $\mathrm{E}[M_k]$ in a secondary segment, and the power-law exponent z that characterizes the distribution of these secondary events. Simulations based on the model accord well with many features of network traffic; in particular, wavelet analysis reveals good agreement with measured first- and second-order statistics for a variety of packet traces.

This model, as well as the fractal Neyman–Scott model considered in the next section (Sec. 13.5.5), are promising candidates for characterizing computer network traffic; we consider both in greater detail in Sec. 13.6.

13.5.5 Fractal Neyman–Scott point process

The final teletraffic model we consider makes use of a fractal Neyman–Scott cluster point process. As with the Bartlett–Lewis process considered above, the primary events derive from a homogeneous Poisson process corresponding to the start times of traffic flows (see Fig. 13.4), but in this model the primary events initiate impulse response functions $h(t)$ with power-law-varying durations (see Prob. 9.2 and Ryu & Lowen, 1995, 1997, 1998). Adding a second Poisson process gives rise to a form of the fractal-shot-noise-driven Poisson process set forth by Lowen & Teich (1991) and studied in Chapter 10. It is a special Neyman–Scott cluster process, as discussed in Sec. 4.5; related models have also been considered by Cox (1984), Mikosch, Resnick, Rootzén & Stegeman (2002), and Latouche & Remiche (2002). The power-law feature of the impulse response function captures the burstiness of the traffic.

In the general case, no direct correspondence is patently obvious between the form of the power-law impulse response function and any particular feature of the network or traffic. For FTP and several other specific forms of traffic, however, file sizes follow a power-law distribution (Paxson & Floyd, 1995; Park et al., 1996; Crovella & Bestavros, 1997). As suggested by Ryu & Lowen (2002), we can therefore make the fractal-shot-noise-driven Poisson process quite realistic by positing a rectangular impulse response function $h(t)$ with a random cutoff time B characterized by a decaying power-law distribution (see Prob. 13.8). This rectangular fractal-shot-noise-driven Poisson model closely mimics observed FTP traffic.

Moreover, this approach permits the use of a queueing representation, as considered in Sec. 13.3.3 (see Fig. B.19, the solution to Prob. 13.8). As discussed in

[11] Inasmuch as the primary and secondary processes *both* give rise to the form of the interevent-interval distribution, its deviation from exponential form should, properly speaking, not be attributed solely to the secondary process.

Sec. 13.4.3, the queue-length histogram is identical to that for a queue comprising Poisson flow arrivals, heavy-tailed service times, and locally Poisson packet arrivals. In particular, the M/G/1/∞ queue represents shot noise with a fixed-height impulse response function $h(t)$. Specifying a power-law form for the service-time distribution G yields fractal shot noise with this fixed-height impulse response function (Likhanov, Tsybakov & Georganas, 1995).

The fractal Neyman–Scott model is similar to, but distinct from, the fractal Bartlett–Lewis model considered in the previous section (Sec. 13.5.4). Since both are promising candidates for characterizing computer network traffic, we examine them in greater detail in the following section (Sec. 13.6).

13.6 IDENTIFYING THE POINT PROCESS

As discussed in Secs. 5.5.4, 11.5.3, and 12.1, identifying a fractal-based point process is not an easy endeavor. In this, the final section of the book, we set forth a step-by-step approach toward identifying an arbitrary fractal-based point process, using computer network traffic as a didactic testbed.

We offer the following steps as a possible blueprint for the analysis of unknown fractal-based point processes.

13.6.1 Compute multiple statistical measures

We begin by gathering a whole range of measures from the experimental point process and presenting them in a single graphic. For the case at hand, Ethernet-traffic data set BC-pOct89, we present nine statistical measures as the solid curves in Fig. 13.7.

While the statistics of the data set itself are vital, calculating the same statistics for surrogate data sets yields further information that is highly valuable for elucidating the nature of the point process. We employ two surrogates. The first, a shuffled surrogate, comprises the same interevent intervals as the original data set, but rearranged into a random order. As described in Sec. 11.5, a shuffled surrogate has marginal interevent-interval statistics that coincide with those of the original data. Since shuffling generally destroys any dependencies among the intervals, thereby rendering them independent, the other statistics mimic those of a renewal point process. In short, shuffling destroys the long-term properties of a data set while preserving its short-term qualities.

The second surrogate achieves essentially the reverse. As described in Sec. 11.4, we construct this surrogate by transforming the interevent intervals from their original form into a specified distribution, while preserving their relative ordering and the mean of the interevent intervals. In particular, we transform the intervals through exponentialization, which yields an exponential interevent-interval density. An exponentialized data set roughly resembles a Poisson process, but with a variable rate.

Figure 13.7a) directly displays the sequence of interevent intervals for the first five seconds of the data set. The average of the two event times flanking an interval forms the abscissa for that interval, while the interval duration determines the ordinate. We express the duration in terms of its mean, $\tau(t)/\widehat{\mathrm{E}}[\tau]$, so that a value of unity corresponds to an interval that equals this average. While no strong pattern emerges from this panel, we see the large variability in the rate, as well as evidence for preferred intervals at and below unity. We do not present shuffled or exponentialized versions of these data since distinguishing among the three types of points would prove difficult.

While the individual intervals betray mainly short-term effects, the longer-term properties of the point process are more readily revealed by variations in the rate, presented in normalized form in panel b). We use a counting time of 0.3 sec to compute the normalized rate, $\lambda_k/\widehat{\mathrm{E}}[\lambda]$, so that the 41 windows shown along the abscissa span 12.3 sec. A value of unity indicates a local rate equal to that of the data set as a whole. The relatively low rate over the first ten or so windows corresponds to the relative preponderance of long intervals in the first three seconds of panel a). As expected, the shuffled surrogate shows far less variability than the original data, since shuffling destroys inter-interval dependencies. The exponentialized surrogate yields results resembling those of the original data.

The interevent-interval histogram displayed in panel c), $\widehat{p}\,(\tau/\widehat{\mathrm{E}}[\tau])$, provides an estimate of the underlying interval probability density (see Sec. 3.3.1). We normalize both the abscissa and ordinate by the mean interevent interval, which yields dimensionless quantities on both axes. The data roughly follow a decaying exponential form, punctuated by a number of peaks. The shuffled surrogate has precisely the same histogram, by construction. The exponentialized form follows a straight line, as it must on this semilogarithmic plot.

The next four panels, d)–g), present interval-based measures that examine dependencies across interevent intervals. All of these measures employ normalization such that independent intervals yield values of unity on the ordinate, and all follow a power-law form (straight line on these plots) for interevent intervals exhibiting fractal correlations.

Panel d) presents the interval-based normalized rescaled range (NR/S), $\widehat{U}_2(k) \equiv \widehat{U}^2(k)/k$, as defined in Secs. 3.3.5 and 12.3.4. This measure follows a power-law form for the original data, indicating fractal behavior. The exponentialized surrogate yields similar results, but lies slightly above the original data. The shuffled surrogate approaches a value of unity, but not closely; the difference stems from the well-known bias inherent in this statistic.

In panel e) we display the normalized detrended fluctuation (NDF), $\widehat{Y}_2(k_2) \equiv 15\,\widehat{Y}^2(k_2)/k_2\mathrm{Var}[\tau]$, where $k_2 = k+2$, as defined in Secs. 3.3.6 and 12.3.5. Rather than plotting the number of intervals on the abscissa, we offset this by two, as explained in Sec. 12.3.5. This measure also exhibits power-law behavior, and perhaps more closely follows the canonical fractal form of Eq. (12.2). Results for the exponentialized data again lie slightly above those of the original data, while the shuffled version yields values quite close to unity, as expected.

We show the normalized interval-based Haar-wavelet variance (NIWV), $\widehat{A}_\tau(k) \equiv \widehat{\mathrm{Var}}[W_{\psi,\tau}(k,l)]/\widehat{\mathrm{Var}}[\tau]$, in panel f). This measure, considered in Secs. 3.3.4 and

12.3.6, yields results quite similar to those of the normalized detrended fluctuation shown in panel e).

Panel g) displays the last of the interval measures, the normalized interval spectrum (NIS), $\widehat{S}_\tau(\mathsf{f})/\widehat{\mathrm{Var}}[\tau]$, considered in Secs. 3.3.3 and 12.3.7. We smoothed this measure as described in Footnote 7 on p. 117. Like the three preceding interval-based measures, the normalized interval spectrum follows a power-law form for both the original and exponentialized data. However, a bump appears in the spectrum at about $\mathsf{f} = 0.05$, corresponding to a conventional frequency $f = \mathsf{f}/\mathrm{E}[\tau] \approx 28$ Hz. The shuffled data fluctuates closely about unity. All three curves approach a value of unity for large frequencies, as imposed by the normalization.

The normalized Haar-wavelet variance (NHWV) shown in panel h), $\widehat{A}(T)$, derives from the sequence of counts rather than from the intervals (see Secs. 3.4.3, 12.2.3, and 12.3.8). The abscissa therefore corresponds precisely to conventional time. All curves achieve a value of unity at small counting times, by construction. For the original data, this measure follows a power-law form for the most part. However, a few bumps appear at shorter times; these are consistent with a periodic component in the neighborhood of 28 Hz, as we also inferred from the interval spectrum in panel g) [see also Prob. 4.10.1 in conjunction with Eq. (3.41)]. In contrast to panels d) through f), the curve for the exponentialized data lies slightly below that of the original data, although it otherwise resembles it in most respects. The shuffled surrogate yields a normalized Haar-wavelet variance that dips slightly below unity at small counting times and then increases to a value of about three at much larger counting times. A renewal point process with an interval coefficient of variation $C_\tau = 1.8$ (see Table 13.2) would yield a similar curve, in accordance with Eq. (4.18). Shuffling retains the interevent-interval statistics, in particular preserving the empirical value $C_\tau = 1.8$ from the original data.

Finally, panel i) presents the normalized rate spectrum (NRS), $\widehat{S}_\lambda(f, T)/\widehat{\mathrm{E}}[\lambda]$, considered in Secs. 3.4.5 and 12.3.9. This measure also has an abscissa that corresponds exactly to a conventional quantity, in this case frequency. To permit the frequency to extend as high as 100 Hz, we employed a Fourier transform of size 2^{19}, the minimum for a data set of this duration. As with the interval-based spectrum in panel g), we smoothed this measure as described in Footnote 7 on p. 117. Normalization by the mean rate forces all curves to attain an asymptote of unity for sufficiently large frequencies; however, the spectrum reaches this asymptote only for frequencies greater than about 10^4 Hz (not shown). (Unnormalized versions of the rate spectrum appear elsewhere — see Footnote 7 on p. 326.) This measure yields results that roughly resemble those for the interval spectrum displayed in panel g), with power-law behavior for both the original and exponentialized data. A peak appears at about 28 Hz, in concert with those observed in panels g) and h). The normalized rate spectrum for the shuffled data achieves a value of about three at low frequencies, in mirror image to the normalized Haar-wavelet variance, as expected on the basis of Eq. (4.17).

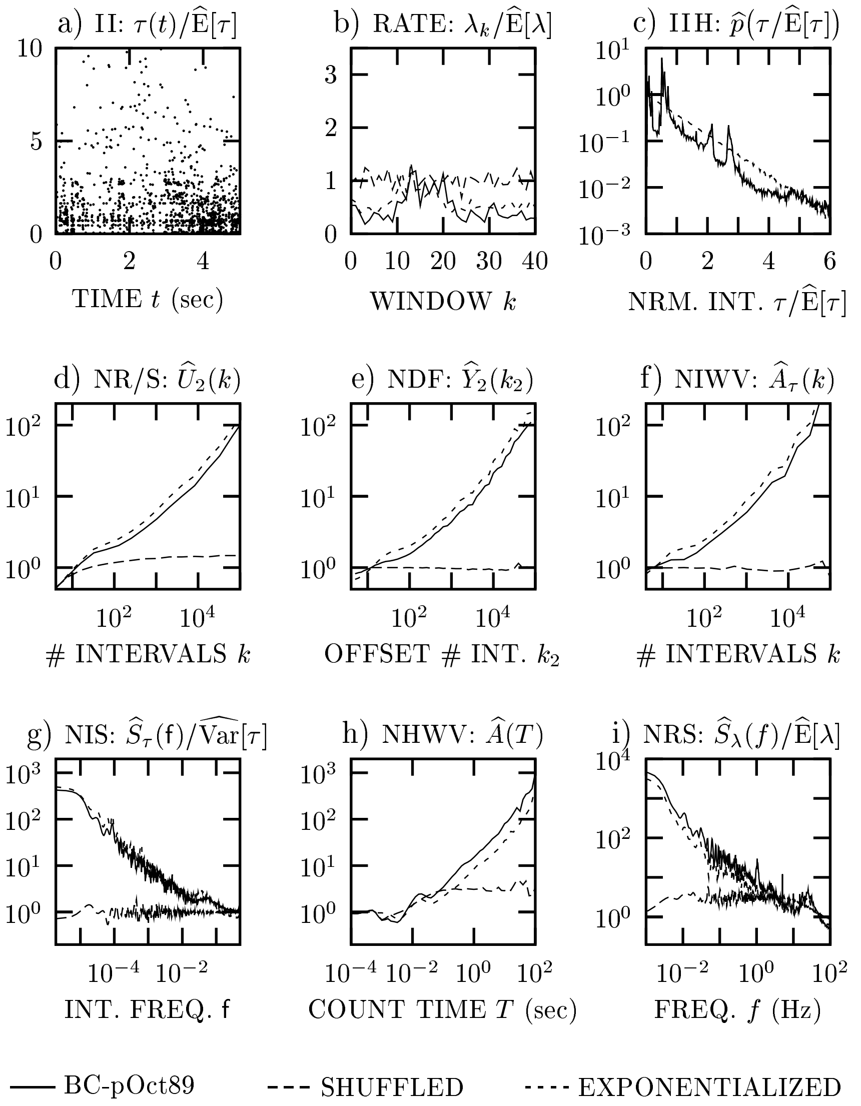

Fig. 13.7 Nine statistical measures for the classic computer network traffic data set BC-pOct89 (solid curves). The data comprise the arrival times of the first 1 million packets recorded on the main Ethernet cable at the Bellcore (BC) Morristown Research and Engineering Facility over a period of some 29 minutes beginning at 11:00 AM on 5 October 1989 (Leland & Wilson, 1989, 1991). Results for the shuffled and exponentialized surrogates appear as the dashed and dotted curves, respectively. We describe the measures and surrogates in the text.

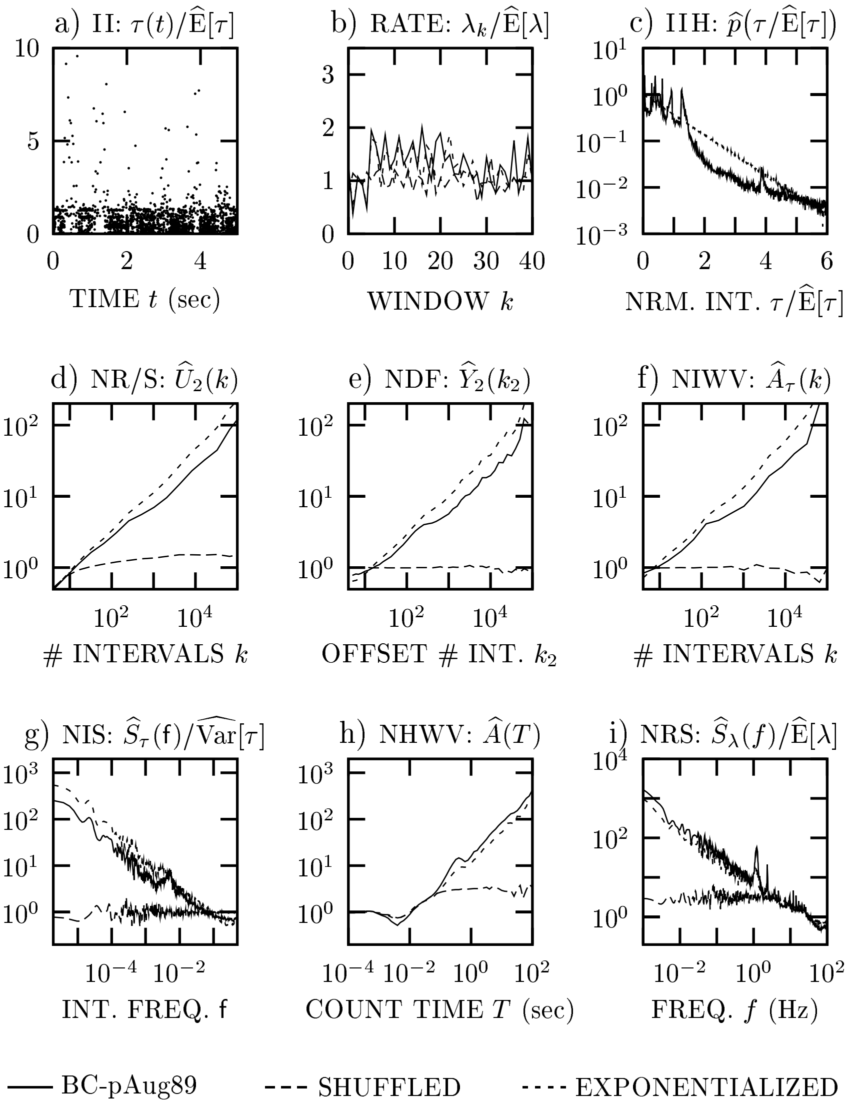

Fig. 13.8 Nine statistical measures for the classic computer network traffic data set BC-pAug89 (solid curves). The data comprise the arrival times of the first 1 million packets recorded on the main Ethernet cable at the Bellcore (BC) Morristown Research and Engineering Facility over a period of some 52 minutes beginning at 11:25 AM on 29 August 1989 (Leland & Wilson, 1989, 1991). Results for the shuffled and exponentialized surrogates appear as the dashed and dotted curves, respectively. The statistics resemble those presented in Fig. 13.7 for data set BC-pOct89.

13.6.2 Compute statistical measures for multiple data sets

Since a similar, but independent, Ethernet-traffic data set is available, we present the same nine statistical measures displayed in Fig. 13.7 as the solid curves in Fig. 13.8. The object of investigating multiple data sets is to establish which features of the data are general characteristics and which appear to vary from one data set to another.

The statistics for BC-pAug89 resemble those for BC-pOct89 in broad outline (compare Figs. 13.7 and 13.8), although they differ in some small details, as we proceed to highlight.

Periodicities are more dominant in the first two panels of Fig. 13.8 than in Fig. 13.7. Panel a) in Fig. 13.8 displays the occurrence of rather long interevent intervals roughly every second or so, the presence of which, in turn, leads to roughly periodic fluctuations of the rate estimate in panel b), based on 0.3-sec counting windows. This periodic component also appears as a peak in the normalized rate spectrum in panel i) at about 1 Hz; a second harmonic also appears.[12] The peaks and valleys in the normalized Haar-wavelet variance displayed in panel h) derive from this as well. However, this strong periodicity is specific to the time at which the data record begins; other times within BC-pAug89 also display this feature, but less strongly. The beginning of this data set also differs from the remainder in that the local rate exceeds the mean. Normalized rates in excess of unity in panel b), and a preponderance of small intervals in panel a), both accord with this observation. Conversely, some portions of data set BC-pOct89 appear more periodic than its beginning. In fact, the periodic component manifested at 28 Hz in BC-pOct89 is totally absent in BC-pAug89.

The interevent-interval histogram displayed in panel c) of Fig. 13.8 reveals different peaks at various normalized interevent times; however, it exhibits an exponential tail in the long-time limit (not shown). Similar behavior appears in panel c) of Fig. 13.7, although the peaks are localized at different interevent times; it, too, exhibits an exponential tail in the long-time limit.

We conclude that data sets BC-pOct89 and BC-pAug89 have similar principal features, although they differ in many details.

13.6.3 Identify characteristic features

Taking the collection of these observations into account, two characteristic features emerge with respect to the data presented in Figs. 13.7 and 13.8: (1) the presence of a fractal rate, and (2) an estimated interevent-interval distribution that does not impart fractal properties to the data. A signature of the first feature is the nearly constant, nonzero slopes of the solid curves in panels d)–i) of both figures. Because shuffling destroys this property (dashed curves), we conclude that the point process belongs to the class of fractal-rate point processes, and not to the class of fractal point processes. This conclusion is confirmed by the behavior of the generalized dimensions D_q for

[12] Since data set BC-pAug89 has a duration that is approximately twice that of BC-pOct89, we doubled the Fourier transform size to 2^{20} to allow the abscissa in panel i) to reach 100 Hz.

these data (see Sec. 3.5.4 and Prob. 5.5), which exhibit integer values (not shown). The capacity-dimension scaling function presented in Fig. 5.10 explicitly illustrates this for D_0.

While estimates of the fractal exponents vary considerably, depending on the measure and the method of estimation (tables not shown), $\widehat{\alpha} = 0.8$ and 0.7 are good compromises for Figs. 13.7 and 13.8, respectively. Obtaining accurate estimates is particularly challenging because of the significant deviations from canonical forms, such as those set forth in Eq. (5.44). As an example, the peaks and valleys near $T = 10^{-2}$ in the estimated normalized Haar-wavelet variance displayed in panel h) of both figures are confounding short-time effects.

As both panels c) show, the exponentialized interevent-interval densities behave as decaying exponential functions (dotted straight lines on these semilogarithmic plots), as they must. The original histograms of both figures (solid curves) exhibit peaks and valleys imparted by preferred intervals in the local traffic flow, and depart significantly from exponential behavior. Furthermore, some slight evidence of a positive curvature exists, particularly near $\tau/\widehat{\mathrm{E}}[\tau] = 4$ and 2, in Figs. 13.7c) and 13.8c), respectively. Power-law curves displayed on a semilogarithmic plot would, in fact, exhibit just such a positive second derivative. However, for the largest intervals shown, the interevent-interval histograms for these data coincide with the exponential form engendered by exponentialization. Indeed, exponential behavior persists at far larger intervals, as demonstrated in Fig. 5.9 for data set BC-pOct89.[13] Thus, while an exponential distribution provides only a fair model of the BC-pOct89 and BC-pAug89 interevent-interval histograms, a power-law distribution would be significantly worse. We conclude that the estimated interevent-interval histograms are nonfractal.

Furthermore, for both Figs. 13.7 and 13.8, the fractal behavior in panels d)–i) (solid curves) is destroyed by shuffling (dashed curves), but modified only slightly by exponentialization (dotted curves), thereby confirming the absence of power-law tails in the interevent-interval histograms. The behavior of these surrogates demonstrates conclusively that the interval distribution does not contribute significantly to the fractal nature of the computer network traffic at hand. Indeed, these observations validate the use of the interval-based measures displayed in panels d)–g) for the analysis of these data, as discussed in Sec. 12.3.1.

13.6.4 Compare with other point processes

Comparing the BC-pOct89 and BC-pAug89 COMPUTER data with the collection of other experimental point-process data examined in Chapter 5 yields a number of interesting parallels and contrasts. Our goal of identifying the point process at hand is also furthered by comparing the COMPUTER surrogates with the surrogates of other

[13] Furthermore, the largest interval exceeds the mean by a factor of ≈ 87 for Fig. 13.7c), and a factor of ≈ 109 for Fig. 13.8c) (see Table 13.2). This is reasonable for 999999 intervals with an exponential tail [$\ln(999999) \approx 14$], but not for a putative power-law distribution that imparts significant fractal behavior to a point process.

experimental point-process data examined in Chapter 11. All of the data sets that we investigated turned out to be fractal-rate point processes. The following comparisons prove useful in identifying the COMPUTER point process:

- The *normalized rate spectra* for the COMPUTER data displayed in Fig. 5.1 [and in Figs. 13.7i) and 13.8i)] reveal spectral features of various widths, sporadically located over a broad range of frequencies. A number of other point processes exhibit similar behavior.

- The *normalized Haar-wavelet variance* curves presented in Fig. 5.2 [and in Figs. 13.7h) and 13.8h)] demonstrate that the fractal exponent of the COMPUTER data has a value $\widehat{\alpha} \approx 0.8$, which is below unity; fractal exponents for the CORTEX, COCHLEA, RETINA, and INTERNEURON (the latter appears in Fig. 11.17) also lie below unity. All of the other data sets have fractal exponents in excess of unity.

- Results gleaned from the corresponding *interval-based spectra,* displayed in Fig. 5.7 [also Figs. 13.7g) and 13.8g)], and *interval-based wavelet variances,* shown in Fig. 5.8 [also Figs. 13.7f) and 13.8f)], offer a broad confirmation of the count-based results reported above. As discussed in Secs. 12.3.1 and 13.6.3, these measures are suitable for use in the analysis of fractal-rate point processes.

  Interval-based measures typically appear smoother than their count-based counterparts. This arises because interval frequency and interval number do not precisely track real frequency and time, respectively. This results in a loss of phase coherence and a concomitant attenuation of narrow local features. The increased smoothness does not signify that interval-based measures are in any way superior to count-based measures, however. In fact, we have already seen in the counting domain that while the normalized variance $\widehat{F}(T)$ appears substantially smoother than the normalized Haar-wavelet variance $\widehat{A}(T)$, the former is significantly inferior to the latter for purposes of estimation (see Fig. 12.8).

- The COMPUTER *interevent-interval histograms* displayed in Fig. 5.9 [and in Figs. 13.7c) and 13.8c)] reveal a number of idiopathic features, of various widths and at sporadic intervals. Several other point processes behave similarly. The COMPUTER interevent-interval histograms most closely resemble those associated with the SYNAPSE and CORTEX.

- For all data sets, the *normalized rate spectra* and *normalized Haar-wavelet variances* for the *randomly deleted surrogates,* shown in Figs. 11.3 and 11.4, respectively, resemble the original curves, displayed in Figs. 5.1 and 5.2, respectively, but the random deletion dilutes the local features.

- For all data sets, the *normalized rate spectra* and *normalized Haar-wavelet variances* for the *shuffled surrogates,* shown in Figs. 11.13 and 11.14, respectively, are devoid of the fractal behavior displayed in Figs. 5.1 and 5.2, indicating that all of the point processes we examined are fractal-rate in nature. Using Eq. (4.18) in conjunction with Figs. 11.14 and 11.17, these surrogates reveal clustered underlying interevent-interval histograms ($C_\tau > 1$) for the COMPUTER, SYNAPSE, CORTEX, and INTERNEURON data, and anticlustered histograms ($C_\tau < 1$) for the HEARTBEAT and COCHLEA data.

- For all data sets, comparison of the *normalized rate spectra* and *normalized Haar-wavelet variances* for the *exponentialized surrogates*, portrayed in Figs. 11.10 and 11.11, respectively, with the corresponding original curves, shown in Figs. 5.1 and 5.2, respectively, reveals a reduction of sharp spectral and temporal features. This follows from the jittering of occurrence times imparted by exponentialization, which results in a loss of phase coherence.

- Taken together, these observations lead us to conclude that the *Ethernet-traffic* COMPUTER *point process* shares an essential similarity with all of the other point processes we have investigated, although it *most closely resembles the striate* CORTEX *point process*.

13.6.5 Formulate and simulate candidate models

In Sec. 13.5 we considered a number of fractal-based point processes as candidate models for computer network traffic. We highlighted the family of cascaded point processes, schematized in Fig. 13.4, since this class of models accommodates the vertical-layer structure of the Internet in a parsimonious way (see Sec. 13.2.3).[14] In particular, we devoted considerable attention to the fractal Bartlett–Lewis point process (Sec. 13.5.4) and the fractal Neyman–Scott point process (Sec. 13.5.5). Although these models are distinct, they nevertheless share many features in common.

In this section we simulate these two point processes using parameters appropriate for the classic Ethernet-traffic data set BC-pOct89 (Leland & Wilson, 1989, 1991). We thereby obtain simulated collections of statistical measures, including surrogates, analogous to those shown in Figs. 13.7 and 13.8. These, in turn, enable us to compare the model results with the original data.

Both simulated processes have primary events $dN_1(t)$ that comprise homogeneous Poisson processes (see Fig. 13.4). The difference in the two cascade models lies in the manner in which the secondary events $dN_{2,k}(t)$ are generated. While an exponential distribution clearly does not do justice to the structure of the interevent-interval histogram displayed in Fig. 13.7c), the accurate modeling of all of its periodicities and favored intervals would require a significant investment in terms of both model adjustment and simulation time. In keeping with the spirit of parsimony discussed in Sec. 13.4.2, we therefore employ a homogeneous Poisson process for the genera-

[14] As emphasized in the solution to Prob. 11.12.4, details regarding the underlying physical or biological phenomena play an important role in framing a proper point-process model.

tion of secondary events as well as for the primary process.[15] This should still yield reasonable results for longer-term effects.

For the Bartlett–Lewis process, fractal behavior in $dN_3(t)$ arises from the power-law distribution of the numbers of events M in each secondary process $dN_{2,k}(t)$. This distribution, together with the mean rates of primaries and secondaries, define the process. Constraints include the mean rate of the process, the fractal onset time, and the fractal exponent. For ease of simulation, we chose the number distribution for the secondary processes to be $\Pr\{M > n\} = (n+1)^{\alpha-1}$ for all $n \geq 0$. For $\alpha = 0.8$, this distribution has a finite mean of approximately $\mathrm{E}[M] \doteq 5.27908$, but infinite variance. Dividing the measured rate of BC-pOct89 (568 sec^{-1}) by $\mathrm{E}[M]$ yields the primary rate, $\mu_1 \doteq 107.595/\text{sec}$, which we round to 110/sec for simplicity. Finally, a secondary rate of $\mu_2 = 160/\text{sec}$ yields approximately correct values for the fractal onset times and frequencies. Note that the secondary rate does not affect the overall rate of the point process, since the M_k events eventually appear in $dN_3(t)$ regardless of this rate. With this combination of values, an average of 3.62937 secondary processes exist at any given time. Table 13.1 provides a summary of the parameters used in carrying out the simulations along with values derived from the models.

For the Neyman–Scott process, we chose a fractal-shot-noise-driven Poisson process with a rectangular impulse-response function $h(K,t)$ of constant height c and varying duration K (see Prob. 9.2.1). With this construct, each secondary process has a constant rate while it exists; both primary and secondary processes are therefore again homogeneous Poisson processes. We chose the same simple power-law distribution for K as we used in Prob. 9.2.2, a generalized Pareto form; this imparts fractal characteristics to the overall point process $dN_3(t)$. This process closely resembles the Bartlett–Lewis point process discussed above. The principal distinction between the two is that the Bartlett–Lewis process specifies the random *number* of events in each secondary process $dN_{2,k}(t)$, whereas the Neyman–Scott process instead specifies its *duration*.

With the form of the secondary-process duration established, there remain four parameters: the primary rate; and c, A, and β from the impulse response function $h(K,t)$. Since the point process has but three constraints (mean rate, fractal exponent, and fractal onset time), a free parameter remains. However, fractal behavior cannot exist below A, since by definition no impulse-response functions exist below this cutoff. If A lies below the fractal onset times, then fractal behavior will be suppressed between the onset time and A, reaching its asymptote at times somewhat larger than A.[16] Since this does not mimic the data at hand, we obviate this problem by choosing

[15] Our model differs from that considered by Hohn et al. (2003) in a number of other respects as well. Ideally, we would obtain results for various candidate secondary point processes $dN_{2,k}(t)$, and then fit the resulting interevent-interval histogram of $dN_3(t)$ (the process as a whole) to the data, adjusting the secondary processes $dN_{2,k}(t)$ as necessary. Invoking the argument of parsimony, we do not carry out this procedure.

[16] Figure B.9 shows a similar effect, in the frequency domain, although it arises from a different origin (random displacement of the events).

	Units	Bartlett- Lewis	Neyman- Scott
Primary Parameters			
Power-Law Exponent α		0.8	0.8
Primary-Process Rate μ_1	(sec^{-1})	110	70
Simulation Duration	(sec)	1760	1760
Secondary-Process Rate μ_2	(sec^{-1})	160	—
Secondary-Process Amplitude c	(sec^{-1})	—	140
Secondary-Process Cutoff A	(msec)	—	10
Derived Expected Values			
Concurrent Secondary Processes		3.63	4.2
Secondary-Process Duration	(msec)	33.0	60.0
Events per Secondary Process		5.28	8.4
Total Aggregate Rate	(sec^{-1})	581	588
Total Number of Events	$(\times 10^6)$	1.02	1.03

Table 13.1 Parameters used for simulating realizations of the Bartlett–Lewis and Neyman–Scott point processes. The entries that apply to both simulations, in the upper portion of the table, derive from the target data set, BC-pOct89; we adjusted the other entries to fit the data (see text for details). The five entries in the lower portion of the table are expected results based on the theoretical properties of the models.

$A = 0.01$ sec, well below estimated fractal onset times. We also fix $\beta = 3 - \alpha = 2.2$, in accord with Eq. (B.196). This determines the average area a of the impulse response function:

$$\mathsf{a} = c\,\mathrm{E}[K] = c\,(\beta - 1)(\beta - 2)^{-1} A = 0.06\,c. \tag{13.14}$$

Together with Eq. (10.10), this fixes the product $\mu_1 c$. Finally, we adjust these two quantities so that they produce a fractal onset time that resembles that of data set BC-pOct89. We used a primary rate $\mu_1 = 70/\text{sec}$, and a height $c = 140/\text{sec}$. Table 13.1 summarizes the values used. With these parameters, an average of 4.2 secondary processes exist at any given time.

With the parameters for both processes established, we simulated the two cascaded point processes. We present collections of simulated statistical measures for the fractal Bartlett–Lewis and fractal Neyman–Scott point processes in Figs. 13.9 and 13.10, respectively. The results exhibited strong sensitivity to the random seed chosen (not shown), as expected for a fractal-based point process.[17] Changing the parameters slightly had the same effect since it effectively shifted the random numbers used

[17] Figure 12.1 displays the effects of using various random seeds in a different context.

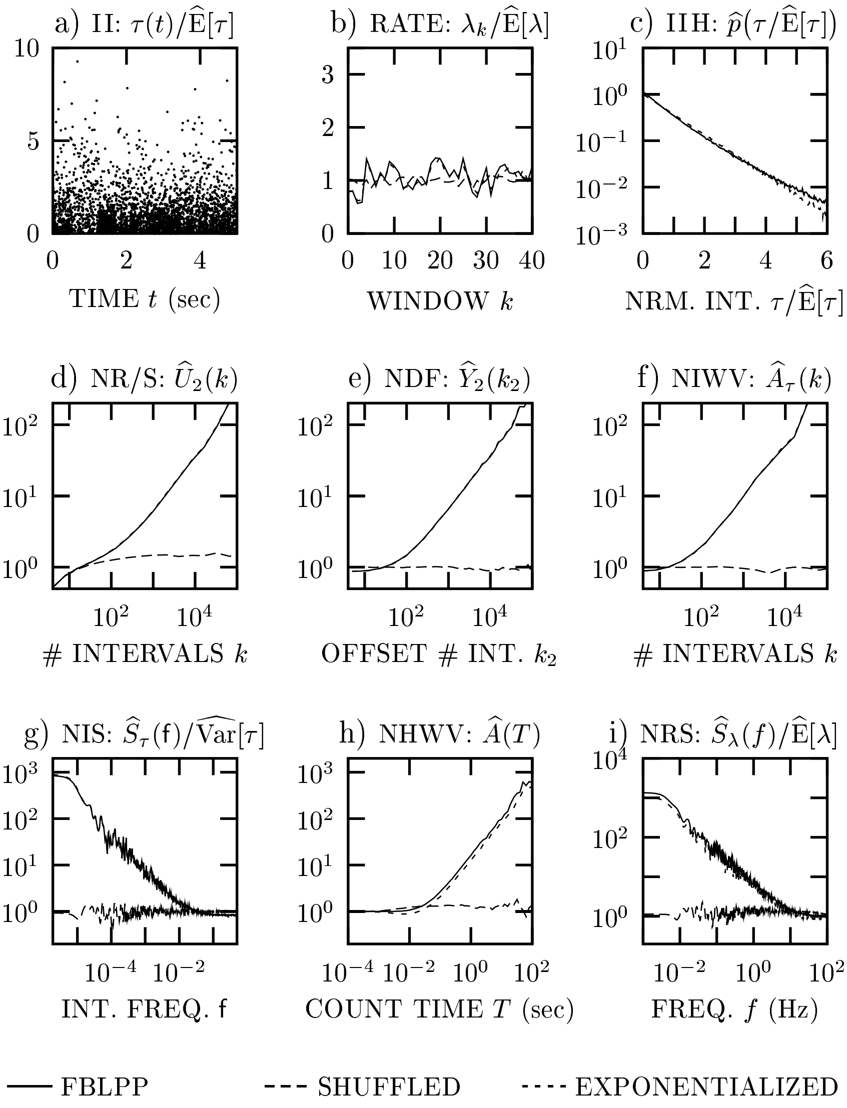

Fig. 13.9 Nine statistical measures for a simulated fractal Bartlett–Lewis point process (FBLPP) with parameters chosen to model the classic computer network traffic data set BC-pOct89 (solid curves). Results for the shuffled and exponentialized surrogates appear as the dashed and dotted curves, respectively. The statistics nicely mimic those shown for BC-pOct89 in Fig. 13.7, particularly those that portray fractal features. The results are quite similar to those generated by the fractal Neyman–Scott point process (see Fig. 13.10).

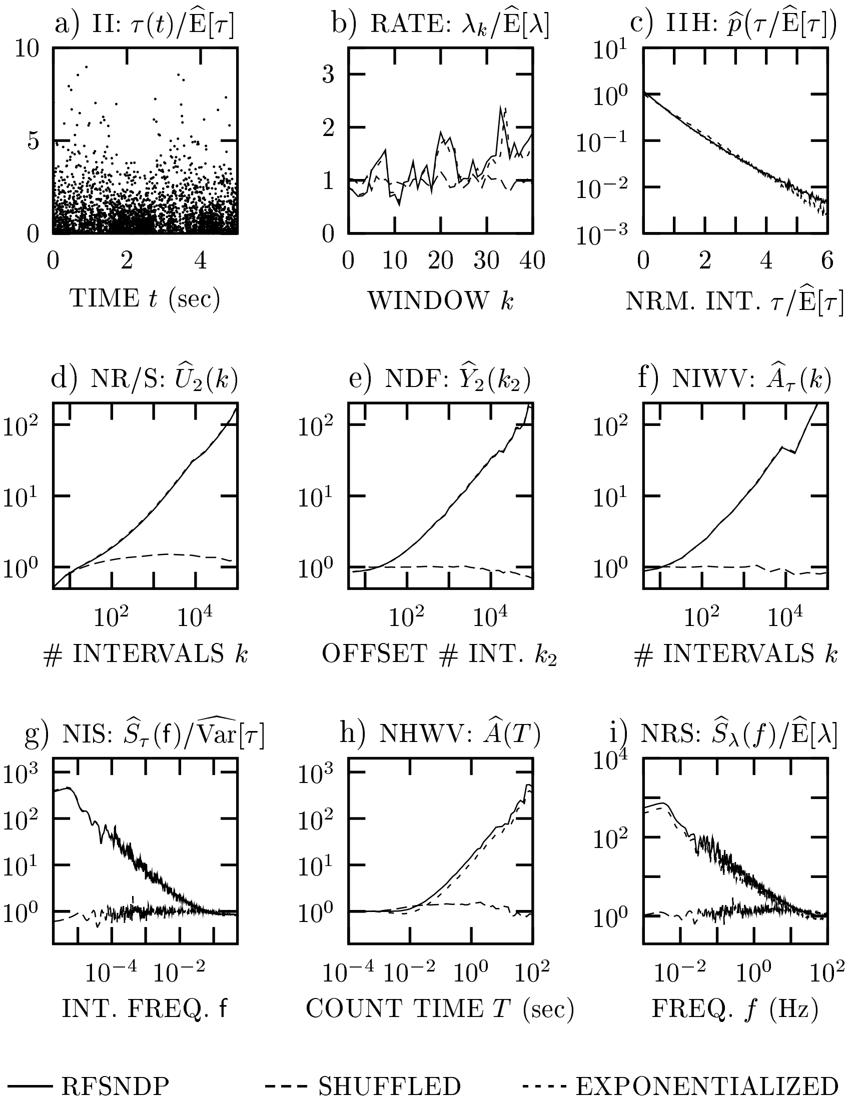

Fig. 13.10 Nine statistical measures for a simulated fractal Neyman–Scott point process (a rectangular fractal-shot-noise-driven Poisson process, RFSNDP) with parameters chosen to model the classic computer network traffic data set BC-pOct89 (solid curves). Results for the shuffled and exponentialized surrogates appear as the dashed and dotted curves, respectively. The statistics mimic those shown for BC-pOct89 in Fig. 13.7 quite well, particularly those that portray fractal features. The results are very similar to those generated by the fractal Bartlett–Lewis point process (see Fig. 13.9).

Interval Statistics	pOct89	pAug89	Bartlett- Lewis	Neyman- Scott
Total Duration L (sec)	1760	3143	1760	1760
Number of Intervals $[N(L) - 1]$	$999\,999$	$999\,999$	$1\,279\,212$	$1\,114\,851$
Minimum Interval (μsec)	16	20	0.000765	0.00108
Maximum Interval (msec)	154	342	36.0	51.3
Mean Interval $\widehat{\mathrm{E}}[\tau]$ (msec)	1.76	3.14	1.38	1.58
Mean Rate $\widehat{\mathrm{E}}[\mu]$ (sec^{-1})	568	318	727	634
Interval Standard Deviation $\widehat{\sigma}_\tau$ (msec)	3.20	5.64	1.59	1.89
Interval Coefficient of Variation $\widehat{C}_\tau$	1.82	1.80	1.16	1.20
Interval Skewness $\widehat{C}_3/\widehat{C}_2^{3/2}$	9.77	9.35	3.07	3.40
Interval Kurtosis $\widehat{C}_4/\widehat{C}_2^2$	170	153	17.5	21.9
Interval Serial Correlation Coefficient $\left\{\widehat{R}_\tau(1) - \widehat{\mathrm{E}}^2[\tau]\right\} \Big/ \widehat{\mathrm{Var}}[\tau]$	0.180	0.200	0.122	0.138

Table 13.2 Representative estimated interval statistics (see Sec. 3.3 for definitions) for two classic Ethernet-traffic data sets: BC-pOct89 and BC-pAug89. The data comprise the arrival times of the first 1 million packets recorded on the main Ethernet cable at the Bellcore (BC) Morristown Research and Engineering Facility on the mornings of 5 October 1989 and 29 August 1989, respectively (see Leland & Wilson, 1989, 1991). Although the mean rate of BC-pOct89 is nearly a factor of two greater than that of BC-pAug89, as shown in row 6, the normalized interval statistics, based on ratios of moments, agree quite closely (see rows 8–11). The skewness, and especially the kurtosis, greatly exceed the values corresponding to a Gaussian distribution, which are zero for the definitions we employ (see Footnote 2 on p. 55). We also include simulated point-process results for two models of computer network traffic: the fractal Bartlett–Lewis cascade point process and the fractal Neyman–Scott cluster point process. The statistics for these two model processes agree with each other quite closely; they also agree reasonably well with the statistics of the two computer network traffic data sets.

among the different stochastic quantities, yielding completely different secondary-point-process durations, for example. On the whole, obtaining precise fits to the target data set, BC-pOct89, proved quite difficult, and certainly beyond any sort of automated minimization procedure such as Marquardt-Levenberg. Rather than search extensively for a particularly lucky set of parameters and random seed, we used round numbers for the parameters, and a default seed. Thus, while we could precisely match the durations of the simulations to that of the BC-pOct89 data set (1759.6 sec), the total number of events varied considerably from the target of $999\,999$; we generated $1\,279\,212$ and $1\,114\,851$ events for the Bartlett–Lewis and Neyman–Scott processes, respectively. Table 13.2 presents a collection of interval statistics derived from these simulations, along with those for the canonical data sets BC-pOct89 and BC-pAug89.

13.6.6 Compare model simulations with data

The similarity in construction between the Bartlett–Lewis and Neyman–Scott cascaded point processes carries forth to their results, which closely resemble each other, as demonstrated in Figs. 13.9 and 13.10, as well as in Table 13.2. By construction, both lack the local bumps, peaks, and valleys evident in the statistics of the computer network traffic data shown in Figs. 13.7 and 13.8 (features such as these also appear in biological point processes, as mentioned in Sec. 13.6.4). As a result, the two model simulations resemble each other a bit more closely than either does the target data set. Lacking explicit mechanisms for generating these effects, both simulations change little with exponentialization.

Aside from this, both simulations share all of the characteristic features of the original data discussed in Sec. 13.6.3. Furthermore, as shown in Table 13.2, the interval statistics for the two model processes agree reasonably well with those of the two computer network traffic data sets, BC-pOct89 and BC-pAug89. Even the interval coefficients of variation for the Bartlett–Lewis and Neyman–Scott simulations, $C_\tau \doteq 1.16$ and 1.20, respectively, are not inordinately different from those of data sets BC-pOct89 and BC-pAug89, $C_\tau \doteq 1.82$ and 1.80, respectively. Interval shuffling has essentially the same effect on the simulations as it does on the original data (compare dashed curves in Figs. 13.7–13.10).

It appears that the Bartlett–Lewis and Neyman–Scott constructs yield statistics that accord quite well with those obtained from the canonical Ethernet-traffic COMPUTER data sets that we studied. Interestingly, of all the point processes we investigated, the COMPUTER point process most closely resembles that observed at the striate CORTEX (see Sec. 13.6.4). In fact, the Neyman–Scott cascade process has also been effectively used to model the sequence of action potentials recorded from striate CORTEX neurons (Teich et al., 1996).

Problems

13.1 M/M/1/∞ *queue-length distribution* How do Eqs. (13.1) and (13.2) change when the buffer size is infinite?

13.2 M/M/M/Q_m *queue-length distribution* Describe the changes required for Eq. (13.1) when M servers handle requests from the same buffer.

13.3 *Buffer overflow probability approximations* For some queueing models with finite-size buffers, the calculation of the buffer overflow probability P_B is mathematically intractable. In such cases one typically solves the relevant equations assuming infinite buffer size (see Prob. 13.1), and then chooses some other representation for buffer overflow. Consider Eq. (13.1) under the simplification of infinite buffer size, and denote the resulting queue-length distribution by $p_\infty(n)$. A number of approximations for the overflow probability are commonly employed, including $p_\infty(Q_m)$, $p_\infty(Q_m + 1)$, $\sum_{n=Qm}^{\infty} p_\infty(n)$, and $\sum_{n=Qm+1}^{\infty} p_\infty(n)$. Derive forms for each of these quantities for the M/M/1 queue and show that for large values of Q_m, the approximate result $p_\infty(Q_m)$ lies closest to the true result $p_Q(Q_m)$.

13.4 M/M/1 *queue buffer design* Consider a homogeneous Poisson process with a mean interevent interval of 10 msec providing an arrival stream to a queue. Suppose that the service times follow an exponential distribution with a mean value of 9 msec and that there is a single server. For this traffic, determine the minimum buffer sizes that give rise to overflow probabilities of no more than 10^{-3}, 10^{-6}, and 10^{-9}.

13.5 M/M/1/∞ *queue simulation* Simulate the queue specified in Prob. 13.4 for 10^6 seconds ($\approx 10^8$ arrivals), but assume now that $Q_m \to \infty$. Plot the estimated queue-length histogram $\widehat{p}_\infty(n)$. To ensure stationarity, include an additional 10^4 seconds at the beginning and discard it. Show that the histogram follows a geometric form. Use this plot in conjunction with the results obtained in Prob. 13.4 to confirm that Eq. (13.12) provides a good estimate for the overflow probability in the M/M/1/Q_m queue.

13.6 *Fractal-Gaussian-process-driven Poisson process queue simulation* Figures 12.1–12.7 and 12.9 provide results derived from simulations of a Poisson process driven by a fractal Gaussian process (FGPDP) with a mean rate $\mathrm{E}[\mu] = 100$, duration $L = 10^4$, fractal exponent $\alpha = 0.8$, onset frequency $f_S = 0.2$, and fractal-Gaussian-process array size $M = 2^{17}$ (of which we used half). We considered this point process in Secs. 6.3.3, 8.4, and 10.6.1, as well as in Chapter 12.

13.6.1. Using this same process, but with L (and M) increased by a factor of 100, simulate the associated G/M/1 queue assuming exponentially distributed service times with a mean value $1/\mu_s = 0.009$. Plot the estimated queue-length histogram for this process on doubly logarithmic coordinates. Compare it with the theoretical M/M/1 result and with a decaying power-law distribution.

13.6.2. Now repeat the simulation, changing the mean service time to $1/\mu_s = 0.005$ while leaving everything else unchanged. Plot the result on semilogarithmic coordinates this time, and explain why it does not follow a fractal form.

13.7 *Shuffled-fractal-process queue simulation* Consider the traffic-process simulation described in Prob. 13.6. Randomly shuffle this simulation and repeat the queueing analysis with an average service time of 0.009. Show that the result agrees well with that obtained for the M/M/1 queue (see Prob. 13.5).

13.8 *Fractal-shot-noise-driven Poisson process queue simulation* Simulate a fractal-shot-noise-driven Poisson process $N(t)$ (see Chapter 10), where the impulse response functions have a rectangular shape of constant height c, and a duration that obeys a decaying power-law distribution (see Prob. 9.2). Specifically, let the probability that the impulse-response-function duration K exceeds a value x take the form

$$\Pr\{K > x\} = \begin{cases} (A/x)^{\beta-1} & x > A \\ 1 & x \le A, \end{cases} \tag{13.15}$$

with $\beta = 2.2$, $A = 1$, and $c = \frac{10}{3}$. Let the primary Poisson process rate $\mu = 5$. This yields an expected interevent interval at the secondary Poisson process of $\mathrm{E}[\tau] = 0.01$. Assume that all impulse response functions are independent and identically distributed. Again, set the duration of the simulation L to 10^6 for an expected number of events $\mathrm{E}[N(L)] = 10^8$; include an additional duration of 10^4 at the beginning of the simulation, and discard these results. Simulate the associated G/M/1 queue using $N(t)$ as the arrival process and assume exponentially distributed service times with a mean value of $1/\mu_s = 0.009$. Plot the estimated queue-length histogram and compare it with the results displayed in Fig. B.16.

13.9 *Modulated-fractal-process queue simulation* We have considered queue-length histograms for several fractal-rate arrival processes (see Figs. B.16, B.17, and B.19). We wish to investigate how a periodically modulated arrival process changes the character of these histograms.

Consider, as a simple example, a Poisson point process $dN_1(t)$ driven by a periodic, deterministic rate of the form

$$\mu(t) = \mu_0[1 + a\cos(\omega_0 t)], \tag{13.16}$$

where a is the modulation depth and we posit $\mu_0/\omega_0 \gg 1$ to ensure that a large number of events occur within each period of the modulated waveform. We can impose such modulation in the following manner: Generate a new point process $dN_3(t)$ from a homogeneous Poisson process $dN_2(t)$ by multiplying the event times of $dN_2(t)$ by a suitable nonlinear function of $\cos(\omega_0 t)$, chosen so that $dN_3(t)$ has the same statistics as $dN_1(t)$. Said differently, we can warp the time axis of the (unmodulated) point process in a periodic manner to generate a result that mimics a sinusoidally modulated inhomogeneous process.

We can impose such periodic time warping on any arbitrary point process. Begin with the fractal-Gaussian-process-driven Poisson process (FGPDP) considered in Prob. 13.6. Carry out the time warping discussed above and generate a point process that mimics a sinusoidally modulated (inhomogeneous) version of the fractal-Gaussian-process-driven Poisson process. Now let the modified point process serve as the arrival process for a G/M/1 queue, which we denote MODULATED-FGPDP/M/1. For the service process, assume exponentially distributed service times with a mean value $1/\mu_s = 0.009$. Use a modulation period $2\pi/\omega_0 = 1$ min and a modulation depth $a = 1$, as defined by Eq. (13.16). Simulate and plot the estimated queue-length histogram for this queueing problem, and compare your result with those obtained in Prob. 13.6.1 (Fig. B.16).

13.10 *Estimating two fractal exponents* Consider a data set that gives rise to a normalized Haar-wavelet variance with two separate power-law regions. The first exhibits a fractal exponent α_1, and extends from T_{A1} to T_{A2}; the second exhibits a fractal exponent $\alpha_2 > \alpha_1$, and extends from T_{A3} to T_{A4}.

13.10.1. The accurate estimation of α_1 and α_2 requires that T_{A2}/T_{A1} and T_{A4}/T_{A3} both exceed 10^3. We also set $T_{A1} = 10\,\mathrm{E}[\tau]$ to ensure a practical process, and require a total duration $L \geq 10\,T_{A4}$ to achieve a reasonably small variance near T_{A4}. How many events must a simulated data set with these properties contain on average?

13.10.2. One can always fit a monofractal form to a bifractal data set. Using the minimum suitable values found in Prob. 13.10.1, and the exponents $\alpha_1 = 0.4$ and $\alpha_2 = 0.8$, calculate the corresponding ideal normalized Haar variance. Plot this bifractal curve, and find the monofractal curve that minimizes the mean-square error on a doubly logarithmic plot. Compare the two curves and comment on the difference, bearing in mind the implications of Eq. (12.25). Repeat this exercise for $T_{A2}/T_{A1} = T_{A4}/T_{A3} = 10$.