Chapter in *The Handbook of Computer Networks*, Hossein Bidgoli (ed.), Wiley, to appear 2007

Network Traffic Modeling

Thomas M. Chen
Southern Methodist University, Dallas, Texas

OUTLINE:

KEY WORDS: traffic model, burstiness, long range dependence, policing, self similarity, stochastic fluid, time series, Poisson process, Markov modulated process, transmission control protocol (TCP).

ABSTRACT

From the viewpoint of a service provider, demands on the network are not entirely predictable. Traffic modeling is the problem of representing our understanding of dynamic demands by stochastic processes. Accurate traffic models are necessary for service providers to properly maintain quality of service. Many traffic models have been developed based on traffic measurement data. This chapter gives an overview of a number of common continuous-time and discrete-time traffic models. Sources are sometimes policed or regulated at the network access, usually by a leaky-bucket algorithm. Access policing can change the shape of source traffic by limiting the peak rate or burstiness. Source traffic may also be regulated by protocol mechanisms such as sliding windows or congestion windows (as in TCP), leading to other traffic models.

INTRODUCTION

Teletraffic theory is the application of mathematics to the measurement, modeling, and control of traffic in telecommunications networks (Willinger and Paxson, 1998). The aim of traffic modeling is to find stochastic processes to represent the behavior of traffic. Working at the Copenhagen Telephone Company in the 1910s, A. K. Erlang famously characterized telephone traffic at the call level by certain probability distributions for arrivals of new calls and their holding times. Erlang applied the traffic models to estimate the telephone switch capacity needed to achieve a given call blocking probability. The Erlang blocking formulas had tremendous practical interest for public carriers because telephone facilities (switching and transmission) involved considerable investments. Over several decades, Erlang's work stimulated the use of queueing theory, and applied probability in general, to engineer the public switched telephone network.

Packet-switched networks started to be deployed on a large scale in the 1970s. Like circuit-switched networks, packet networks are designed to handle a certain traffic capacity. Greater network capacity leads to better network performance and user satisfaction, but requires more investment by service providers. The network capacity is typically chosen to provide a target level of quality of service (QoS). QoS is the network performance seen by a packet flow, measured mainly in terms of end-to-end packet loss probability, maximum packet delay, and delay jitter or variation (Firoiu, et al., 2002). The target QoS is derived from the requirements of applications. For example, a real-time application can tolerate end-to-end packet delays up to a maximum bound.

Unfortunately, teletraffic theory from traditional circuit-switched networks could not be applied directly to emerging packet-switched networks for a number of reasons. First, voice traffic is fairly consistent from call to call, and the aggregate behavior of telephone users does not vary much over time. However, packet networks carry more diverse data traffic, for example, e-mail, file transfers, remote login, and client-server transactions. Data applications are typically "bursty" (highly variable over time) and vary in behavior from each other. Also, the traffic diversity has increased with the growing number of multimedia applications. Second, traffic is controlled differently in circuit- and packet-switched networks. A circuit-switched telephone call proceeds at a constant bit-rate after it is accepted by the network. It consumes a fixed amount of bandwidth at each telephone switch. In contrast, packet flows may be subjected to access control (rate enforcement at the network boundary); flow control (the destination slowing down the sender); congestion control (the network slowing down the sender); and contention within the network from other packet flows. Thus, packet traffic exhibits much more complex behavior than circuit-switched voice.

Teletraffic theory for packet networks has seen considerable progress in recent decades (Adas, 1997; Frost and Melamed, 1994; Michiel and Laevens, 1997; Park and Willinger, 2000). Significant advances have been made in long-range dependence, wavelet, and multifractal approaches. At the same time, traffic modeling continues to be challenged by evolving network technologies and new multimedia applications. For example, wireless technologies allow greater mobility of users. Mobility must be an additional consideration for modeling traffic in wireless networks (Thajchayapong and Peha, 2006; Wu, Lin, and Lan, 2002). Traffic modeling is clearly an ongoing process without a real end. Traffic models represent our best current understanding of traffic behavior, but our understanding will change and grow over time.

This chapter presents an overview of commonly used traffic models reflecting recent developments in the field. The first step in modeling is understanding the statistical characteristics of the traffic. The first section reviews common statistical measures such as burstiness, long range dependence, and frequency analysis. Next, we examine common continuous-time and discrete-time source models. These models are sufficiently general to apply to any type of traffic, but application-specific models are tailored closer to particular applications. We highlight studies of the major Internet applications: World Wide Web, peer-to-peer file sharing, and streaming video. In the last section, we examine two major ways that the network affects the source traffic. First, traffic sources can be “policed” at the network access.

Second, the dynamics of TCP flows (the majority of Internet traffic) are affected by network congestion and active queue management schemes at network nodes.

Packets, flows, and sessions

Some terminology should be introduced at this point because traffic can be viewed at different levels, as shown in Figure 1. When the need arises, a host will establish a session with another host. A session is associated with a human activity. For example, a client host will open a TCP connection to port 21 on a server to initiate a FTP session. The TCP connection will be closed at the end of the FTP session. Or a session may be viewed as the time interval when a dial-up user is connected to an ISP. For connection-oriented networks such as ATM, a session is a call established and terminated by signaling messages. Traffic modeling at the session (or call) level consists of characterizing the start times and duration of each session.

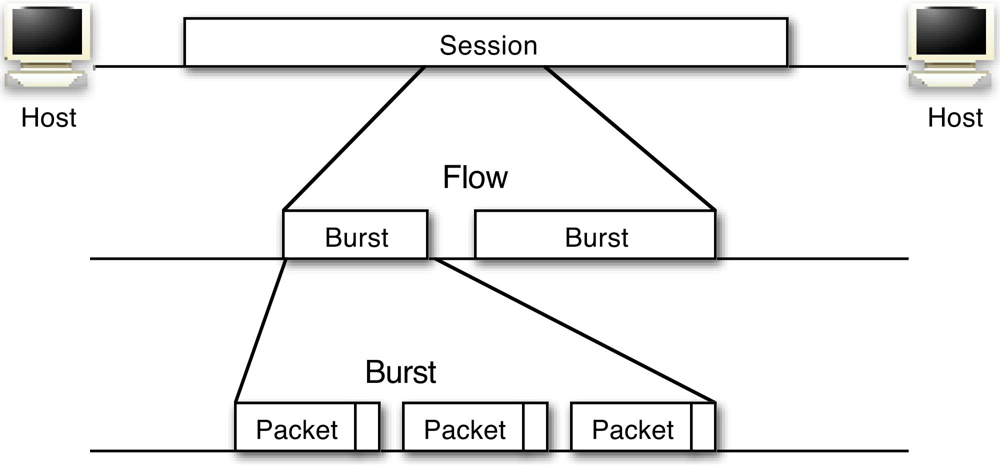

Fig. 1. Levels of traffic

During a session, each host may transmit one or more packet flows to the other host (Roberts, 2004). Although the term is used inconsistently in the literature, a flow is commonly considered to be a series of closely spaced packets in one direction between a specific pair of hosts. Packets in a flow usually have common packet header fields such as protocol ID and port numbers (in addition to source and destination addresses). For example, an FTP session involves two packet flows between a client and server: one flow is the control stream through TCP port 21, and the second flow is the data stream (through a negotiated TCP port). Traffic modeling at the flow level consists of characterizing the random start times and durations of each flow.

TCP flows have been called "elephants" and "mice" depending on their size. An elephant's duration is longer than the TCP slow start phase (the initial rate increase of a TCP connection until the first dropped packet). Due to their short duration, mice are subject to TCP's slow start but not to TCP's congestion avoidance algorithm. Less common terms are "dragonflies," flows shorter than two seconds, and "tortoises," flows longer than 15 minutes (Brownlee and Claffy, 2003). Traffic measurements have suggested that 40-70 percent of Internet traffic consist of short flows, predominantly Web traffic. Long flows (mainly non-Web traffic) are a minority of the overall traffic, but have a significant effect because they can last hours to days.

Viewed in more detail, a flow may be made up of intermittent bursts, each burst consisting of consecutively transmitted packets. Bursts may arise in window-based protocols where a host is allowed to send a window of packets, then must wait to receive credit to send another window. Another example is an FTP session where a burst could result from each file transferred. If a file is large, it will be segmented into multiple packets. A third example is a talkspurt in packet voice. In normal conversations, a person alternates between speaking and listening. An interval of continuous talking is a talkspurt, which results in a burst of consecutive packets.

Finally, traffic can be viewed at the level of individual packets. This level is concerned only with the arrival process of packets and ignores any higher structure in the traffic (bursts, flows, sessions). The majority of research (and this chapter) address traffic models mainly at the packet level. Studies at the packet level are relatively straightforward because packets can be easily captured for minutes or hours.

Studies of traffic flows and sessions require collection and analysis of greater traffic volumes because flows and sessions change over minutes to hours, so hours or days of traffic need to be examined. For example, one analysis involved a few packet traces of several hours each (Barakat, Thiran, Iannaccone, Diot, and Owezarski, 2003). Another study collected eight days of traffic (Brownlee and Claffy, 2002).

The modeling process

Traffic models reflect our best knowledge of traffic behavior. Traffic is easier to characterize at sources than within the network because flows of traffic mix together randomly within the network. When flows contend for limited bandwidth and buffer space, their interactions can be complex to model. The "shape" of a traffic flow can change unpredictably as the flow progresses along its route. On the other hand, source traffic depends only on the rate of data generated by a host independent of other sources.

Our knowledge is gained primarily from traces (measurements) of past traffic consisting of packet arrival times, packet header fields, and packet sizes. A publicly available trace of Ethernet traffic recorded in 1989 is shown in Figure 2. A trace of *Star Wars IV* encoded by

MPEG-4 at medium quality is shown in Figure 3 (data rate per frame). Traffic can be measured by packet sniffers or protocol analyzers, which are specialized pieces of hardware and software for recording packets at link rates (Thompson, et al., 1997). Common sniffers are tcpdump, Snort, and Ethereal. These can be easily connected to transmission links, local area networks, or mirrored ports on switches and routers. Also, traffic volume is routinely recorded by routers as part of the simple network management protocol (SNMP). In addition, some NetFlow-capable routers are able to record simple flow information (e.g., flow start/stop times, number of bytes, and number of packets).

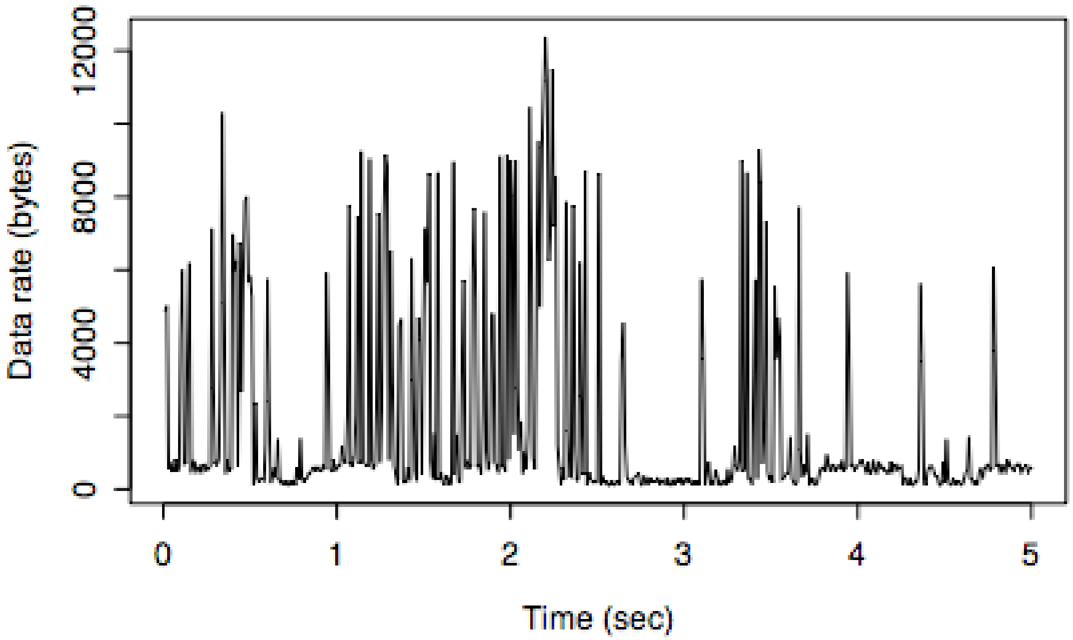

Fig. 2. Ethernet traffic trace.

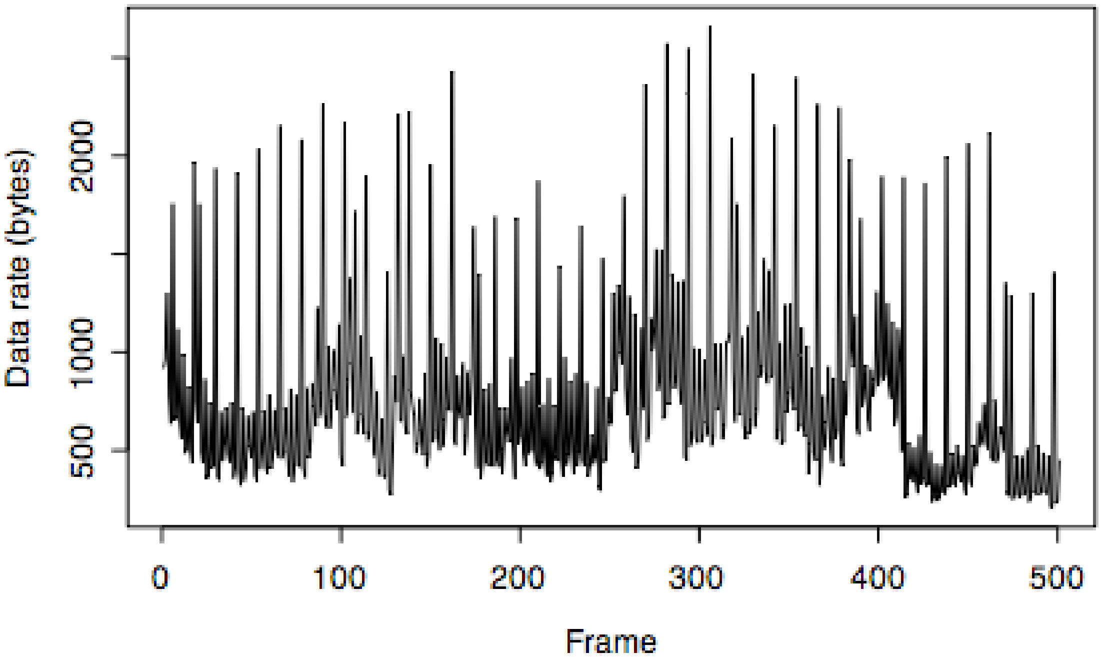

Fig. 3. Trace of *Star Wars IV* encoded at medium quality MPEG-4.

One of the practical difficulties in traffic modeling is collecting and analyzing large sets of traffic measurements. Traffic can be highly variable, even between two similar types of sources. For example, one video might have many scene changes reflected by frequent spikes in the source rate, while another video might have few scene changes resulting in a smoother source rate. Therefore, it is good practice to collect measurements from many sources or over many time periods, and then look for common aspects in their behavior. In probabilistic terms, each traffic trace is a single realization or sample of the traffic behavior. A small sample set could give a misleading portrayal of the true traffic behavior.

A good choice for stochastic model should exhibit accuracy and universality. Accuracy refers to a close fit between the model and actual traffic traces in statistical terms. Sometimes accuracy is judged by the usefulness of a model to predict future behavior of a traffic source. Universality refers to the suitability of a model for a wide range of sources. For example, an MPEG traffic model should be equally applicable to any particular video source, not just to *Star Wars* (although the parameter values of the model may need to change to fit different sources).

Our knowledge of traffic is augmented from analyses and simulations of protocols. For example, we know that TCP sources are limited by the TCP congestion avoidance algorithm. The TCP congestion avoidance algorithm has been analyzed extensively, and TCP sources have been simulated to investigate the interactions between multiple TCP flows to verify the stability and fairness of the algorithm.

Ultimately, a traffic model is a mathematical approximation for real traffic behavior. There are typically more than one possible model for the same traffic source, and the choice depends somewhat on subjective judgment. The choice often lies between simple, less accurate models and more complex, accurate models. An ideal traffic model would be both simple to use and accurate, but there is usually a trade-off between simplicity and accuracy. For example, a common time series model is the p-order autoregressive model (discussed later). As the order p increases, the autoregressive model can fit any data more accurately, but its complexity increases with the number of model parameters.

Uses of traffic models

Given the capability to capture traffic traces, a natural question is why traffic models are needed. Would not traffic measurements be sufficient to design, control, and manage networks?

Indeed, measurements are useful and necessary for verifying the actual network performance. However, measurements do not have the level of abstraction that makes traffic models useful. Traffic models can be used for hypothetical problem solving whereas traffic measurements only reflect current reality. In probabilistic terms, a traffic trace is a realization of a random process, whereas a traffic model is a random process. Thus, traffic models have universality. A traffic trace gives insight about a particular traffic source, but a traffic model gives insight about all traffic sources of that type.

Traffic models have many uses, but at least three major ones. One important use of traffic models is to properly dimension network resources for a target level of QoS. It was mentioned earlier that Erlang developed models of voice calls to estimate telephone switch capacity to achieve a target call blocking probability. Similarly, models of packet traffic are needed to estimate the bandwidth and buffer resources to provide acceptable packet delays and packet loss probability. Knowledge of the average traffic rate is not sufficient. It is known from queueing theory that queue lengths increase with the variability of traffic (Kleinrock, 1976). Hence, an understanding of traffic burstiness or variability is needed to determine sufficient buffer sizes at nodes and link capacities (Barakat, et al., 2003).

A second important use of traffic models is to verify network performance under specific traffic controls. For example, given a packet scheduling algorithm, it would be possible to evaluate the network performance resulting from different traffic scenarios. For another example, a popular area of research is new improvements to the TCP congestion avoidance algorithm. It is critical that any algorithm is stable and allows multiple hosts to share bandwidth fairly, while sustaining a high throughput. Effective evaluation of the stability, fairness, and throughput of new algorithms would not be possible without realistic source models.

A third important use of traffic models is admission control. In particular, connection-oriented networks such as ATM depend on admission control to block new connections to maintain QoS guarantees. A simple admission strategy could be based on the peak rate of a new connection; a new connection is admitted if the available bandwidth is greater than the peak rate. However, that strategy would be overly conservative because a variable bit-rate connection may need significantly less bandwidth than its peak rate. A more sophisticated admission strategy is based on effective bandwidths (Kelly, 1996). The source traffic behavior is translated into an effective bandwidth between the peak rate and average rate, which is the specific amount of bandwidth required to meet a given QoS constraint. The effective bandwidth depends on the variability of the source.

SOURCE TRAFFIC STATISTICS

This section gives an overview of general statistics to characterize source traffic. The traffic may be represented by its time-varying rate $X(t)$ in continuous time or X_n in discrete time. In practice, we are often measuring the amount of data generated over short periodic time intervals, and then the traffic rate is the discrete-time process X_n. A discrete-time process is also easier for computers to record and analyze (as a numerical vector) than a continuous-time process. On the other hand, it might be preferable to view the traffic rate as a continuous-time process $X(t)$ for analysis purposes, because the analysis in continuous time can be more elegant. For example, video source rates are usually characterized by the data per frame, so X_n would represent the amount of data generated for the *n*th frame. But X_n could be closely approximated by a suitable continuous-time process $X(t)$ for convenience, if analysis is easier in continuous time. Alternatively, traffic may be viewed as a point process defined by a set of packet arrival times $\{t_1, t_2, \ldots\}$ or equivalently a set of interarrival times $\tau_n = t_n - t_{n-1}$.

When a traffic trace is examined, one of the first analysis steps is calculation of various statistics. Statistics can suggest an appropriate traffic model. For example, an exponential autocorrelation function is suggestive of a first-order autoregressive process (discussed later).

<u>Simple statistics</u>

The traffic rate is usually assumed to be a stationary or at least wide sense stationary (WSS) process for convenience, although stationarity of observed traffic is impossible to prove because it is a property over an infinite (and unobservable) time horizon. WSS means that the mean $E(X_t)$ is constant over all time *t*, and the autocovariance function

$$R_X(t,s) = E[(X_t - E(X_t))(X_s - E(X_s))] \qquad (1)$$

depends strictly on the lag $|t-s|$ and can be written as $R_X(t-s)$.

First-order statistics such as peak rate and mean rate are easy to measure. The marginal probability distribution function can be estimated by a histogram. Second-order statistics include the variance and autocorrelation function

$$\rho_X(t-s) = R_X(t-s) / R_X(0). \qquad (2)$$

An example of the autocorrelation function estimated for the Ethernet traffic from Figure 2 is shown in Figure 4 (lag is in units of 10 ms). The variance indicates the degree of variability in the source rate, while the autocorrelation function gives an indication of the persistence of bursts (long range dependence is discussed later). The shape of the autocorrelation function is informative about the choice of a suitable time series model (discussed later). Equivalently, the power spectral density function, which is the Fourier transform of the autocorrelation function, is just as informative.

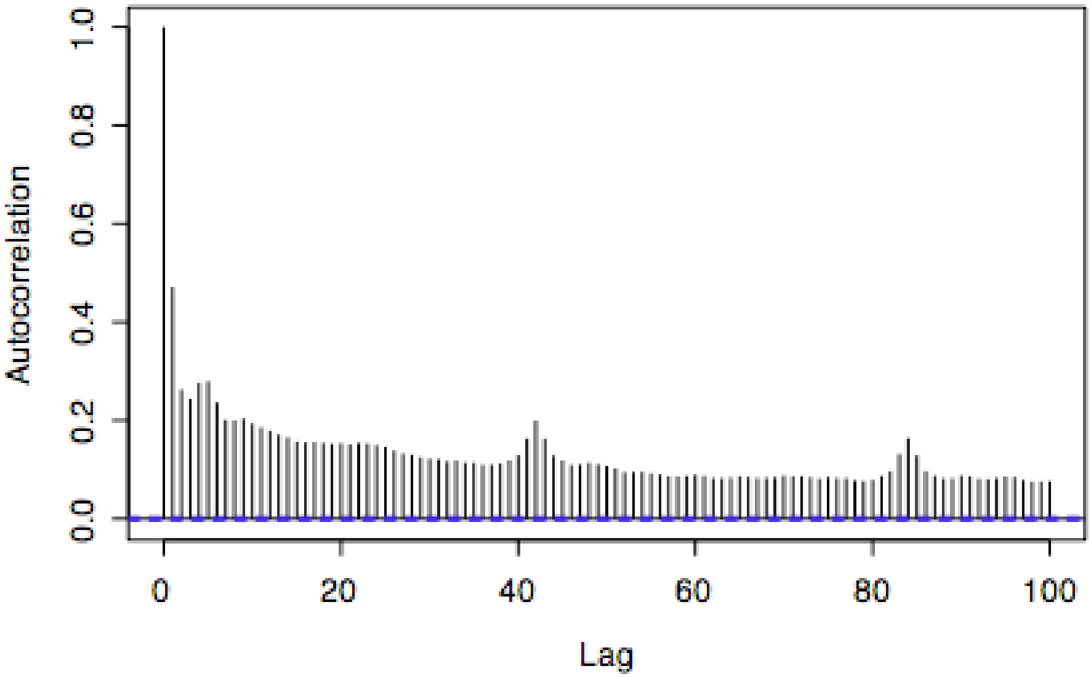

Fig. 4. Autocorrelation function for Ethernet traffic.

Burstiness measures

For variable bit-rate sources, "burstiness" is a quantity reflecting its variability. While burstiness is easy to understand intuitively, there is unfortunately not a universally agreed definition of burstiness. The simplest definition of burstiness is the ratio of peak rate to mean rate. This does not reveal much information about the source because it does not capture how long the peak rate can be sustained (which would have the greatest effect on queueing).

Another possibility is the squared coefficient of variation, treating the traffic rate as samples of a random variable X. The squared coefficient of variation $c^2(X) = \mathrm{var}(X) / E^2(X)$ is a normalized version of the variance of X (normalized by dividing by the squared mean).

A related but more sophisticated measure is the index of dispersion of counts or IDC, $I_c(t) = \mathrm{var}(N(t)) / E(N(t))$, the variance of the number of arrivals up to time t normalized by the mean number of arrivals (Gusella, 1991). The number of arrivals up to time t, $N(t)$, is the number of interarrival times fitting in the interval (0,t). Thus, the index of dispersion of counts captures similar but more timescale-dependent information than the coefficient of variation $c^2(X)$.

Perhaps a more accurate measure is the index of dispersion of workload (IDW), $I_w(t) = \mathrm{var}(Y(t)) / E(Y(t))$, where $Y(t)$ is the total amount of data that has arrived up to time t. The workload takes into account the amount of data in packets, not only the number of packets (Fendick, et al., 1991).

Long range dependence and self similarity

Measurements of Ethernet, MPEG video, and World Wide Web traffic have pointed out the importance of self similarity in network traffic (Beran, 1994; Crovella and Bestavros, 1997; Erramilli, Roughan, Veitch, and Willinger, 2002; Leland, et al., 1994; Paxson and Floyd, 1995; Sahinoglu and Tekinay, 1999). Suppose the traffic rate is a discrete-time WSS process X_n with mean $\overline{X}$ and autocorrelation function $\rho_X(k)$. For positive integer m, a new process $X_n^{(m)}$ is constructed as shown below by averaging the original process over non-overlapping blocks of size m:

$$X_n^{(m)} = (X_{nm} + \cdots + X_{nm+m-1}) / m \,. \qquad (3)$$

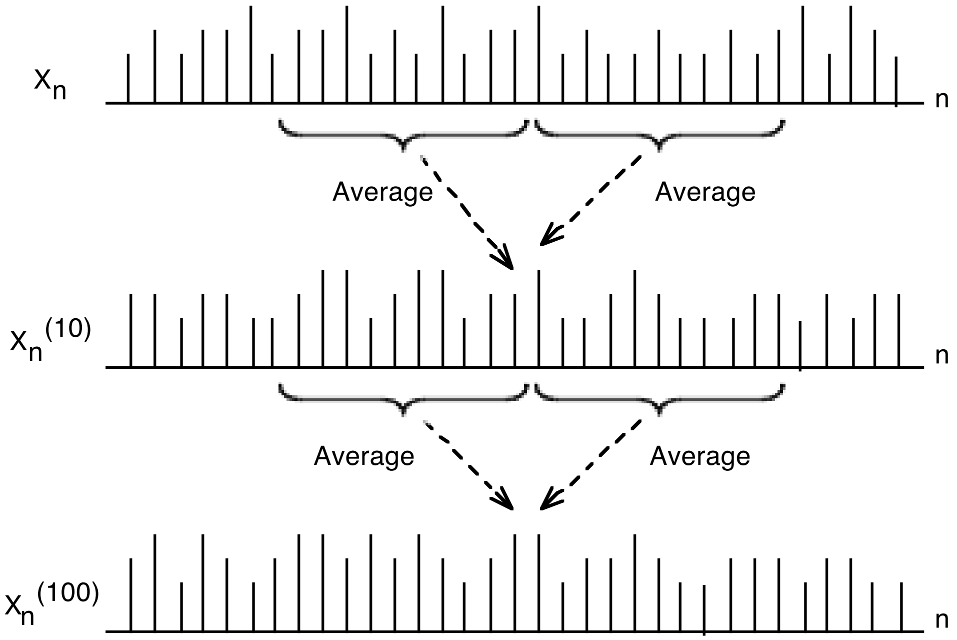

Fig. 5. Construction of rescaled process by averaging over blocks.

The new process $X_n^{(m)}$ is WSS with mean $\bar{X}$ and autocorrelation function $\rho_X{}^{(m)}(k)$. The process X_n is asymptotically second-order self-similar if $\rho_X{}^{(m)}(k) = \rho_X(k)$ for large *m* and *k*. Essentially, self similarity means that the rescaled process $a^{-H}X_{an}$ has the same statistical properties as the original process X_n over a wide range of scales $a > 0$ and some parameter $0 < H < 1$. This would not be true for a process such as the Poisson process, which tends to become smoother for large *m*. The Hurst parameter *H* is a measure of self similarity.

Self similarity is manifested by a slowly decaying autocorrelation function:

$$\rho_X(k) \sim ck^{2(H-1)} \tag{4}$$

for some $c > 0$ as $k \to \infty$. Self similarity implies long range dependence which is a property of the asymptotic rate of decay of the autocorrelation function. A short range dependent (SRD) process has an autocorrelation function that decays at least as fast as exponentially:

$$\rho_X(k) \sim \gamma^{-k} \qquad (5)$$

for large k and some constant γ. Equivalently, the sum $\sum_k \rho_X(k)$ is finite. A long range dependent (LRD) process has an autocorrelation function that decays slower than exponentially, e.g., hyperbolically $\rho_k(k) \to k^{2(H-1)}$ for $0.5 < H < 1$. In this case, the sum $\sum_k \rho_X(k)$ is infinite.

Self similarity and long range dependence are important statistical characteristics because they believed to have a major effect on queueing performance (Erramilli, Roughan, Veitch, and Willinger, 2002). Essentially, bursts of LRD traffic tends to be more persistent, thereby causing longer queue lengths than estimated for traditional SRD traffic such as Poisson.

Multiresolution timescale

The establishment of long range dependence in network traffic prompted interest in the analysis of self-similar processes using timescale methods, particularly wavelets (Abry and Veitch, 1998; Ma and Ji, 2001). Wavelet transforms are a useful tool for examining processes with scaling behavior, referred to as multiresolution analysis. They are most easily approached and motivated from the familiar Fourier transform which decomposes a process (or signal) into a weighted sum of sinusoids. That is, the Fourier transform gives information about the frequency components but not any time information (where the frequency components occur in time). This is well suited for stationary processes but not well suited for non-stationary processes. In contrast, the wavelet transform gives both time-frequency information. The short-time Fourier transform can handle non-stationary processes by taking windows of the signal, but there is a trade-off between time and frequency resolution.

A continuous wavelet transform decomposes a process into a weighted sum of wavelet functions. The wavelet functions are derived from a "mother wavelet" $\psi(t)$, for example, the Haar wavelet, Morlet wavelet, Mexican Hat, or Daubechies family of wavelets. The set of wavelets is generated by dilating the mother wavelet,

$$\psi_s(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t}{s}\right) \tag{6}$$

where *s* is the scale parameter. Larger scales correspond to dilated signals and give a global view of the process (low frequencies), while the opposite holds for smaller scales. The continuous wavelet transform of a process *X(t)* is

$$CWT(b,s) = \frac{1}{\sqrt{|s|}}\int_{-\infty}^{\infty} X(t)\psi\left(\frac{t-b}{s}\right)dt \tag{7}$$

where *b* is the translation parameter.

In order to facilitate computer processing, it is necessary to discretize the transform by sampling the time-frequency plane. The discrete wavelet transform analyzes a process at dyadic scales using inner products at shifts equal to the length of the wavelet. It is computed by passing the process through a series or bank of filters. The first level consists of parallel low-pass and high-pass filters, followed by downsampling by two. The level 1 coefficients are output from the downsampled high-pass filter. The output of the low-pass filter goes again through another filter stage (with high and low pass filters, and downsampling) to increase the frequency resolution. The level 2 coefficients are output from the high-pass filter in the second stage, and additional stages of filters can be used in the same manner. The result is a set of coefficientst that represent the process as a low resolution approximation and additional approximations of increasing resolutions.

An early use of the DWT was estimation of the Hurst parameter (Abry and Veitch, 1998; Roughan, Veitch, and Abry, 2000). Wavelet transforms have been been used to analyze video and Ethernet traffic for the purpose of accurately synthesizing traffic with both long and short range dependence (Ma and Ji, 2001).

Scaling

Recent studies of network traffic have found complex scaling behavior at small timescales (Riedi, et al., 1999; Gilbert, Willinger, and Feldmann, 1999; Abry, et al., 2002). Previous studies had focused on monofractal models, such as self-similar processes. These show invariance across a wide range of scales and can be characterized by a single scaling law, for example, the Hurst parameter *H*.

Multifractal models have been found to better explain scaling behavior at small time scales. Loosely speaking, multifractal models have local scaling properties dependent on a time-varying $h(t)$, the local Hölder exponent, instead of a constant Hurst parameter. Intervals where $h(t) < 1$ show bursts while intervals where $h(t) > 1$ correspond to small fluctuations. In traffic measurements, the local scaling $h(t)$ appears to change randomly in time.

CONTINUOUS-TIME SOURCE MODELS

Traffic models can be continuous-time or discrete-time. In continuous time, we are interested in a stochastic process to represent the time-varying source rate $X(t)$ or the set of packet arrival times $\{t_1, t_2, \ldots\}$.

Traditional Poisson process

In traditional queueing theory, the Poisson arrival process has been a favorite traffic model for data and voice. The traditional assumption of Poisson arrivals has been often justified by arguing that the aggregation of many independent and identically distributed renewal processes tends to a Poisson process when the number increases (Sriram and Whitt, 1986). Poisson arrivals with mean rate λ are separated by interarrival times $\{\tau_1, \tau_2, \ldots\}$ that are i.i.d. (independent and identically distributed) according to an exponential probability distribution function $p(\tau) = \lambda e^{-\lambda\tau}$, $\tau \geq 0$. Equivalently, the number of arrivals up to time *t* is a Markov birth process with all birth rates of λ .

The Poisson arrival process has several properties that make it appealing for analysis.

- It is memoryless in the sense that, given the previous arrival occurred *T* time ago, the time to the next arrival will be exponentially distributed with mean $1/\lambda$ regardless of *T*. In other words, the waiting time for the next arrival is independent of the time of the previous arrival. This memoryless property simplifies analysis because future arrivals do not need to take into account the past history of the arrival process.
- The number of arrivals in any interval of length *t* will have a Poisson probability distribution with mean λt .
- The sum of two independent Poisson arrival processes with rates λ_1 and λ_2 , is a Poisson process with rate $\lambda_1 + \lambda_2$. This is convenient for analysis because traffic flows are multiplexed in a network.
- Suppose a Poisson arrival process with rate λ is randomly split, meaning that each arrival is diverted to a first process with probability *p* or a second process with probability 1-*p*. The first process is a Poisson arrival process with rate $p\lambda$ and the second process is another Poisson process with rate $(1-p)\lambda$.

Despite the attractiveness of the Poisson model, its validity for real traffic has been often questioned. Barakat, et al. (2003) offered evidence that flow arrivals on Internet backbone links are well matched by a Poisson process.

For large populations where each user is independently contributing a small portion of the overall traffic, user sessions can be assumed to follow a Poisson arrival process (Roberts, 2004). Based on traces of wide-area TCP traffic, Poisson arrivals appears to be suitable for traffic at the session level when sessions are human initiated, e.g., interactive TELNET and FTP sessions (Paxson and Floyd, 1995). However, the Poisson model does not hold for machine-initiated sessions nor any packet-level traffic.

<u>Simple on/off model</u>

The simple on/off model is motivated by sources that alternate between active and idle periods. An obvious example is speech where a person is either talking or listening. The time in the active state is exponentially distributed with mean $1/\alpha$, and the time in the idle state is exponentially distributed with mean $1/\beta$. The state of the source can be represented by the two-state continuous-time Markov chain shown in Figure 6.

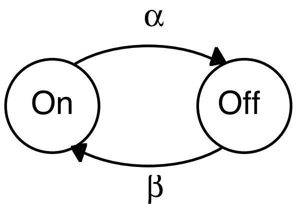

Fig. 6. Two-state Markov chain

In the active state, the source is generating packets at a constant rate 1/*T*, i.e., interarrival times between packets are a constant *T*. In the idle state, no packets are generated. This on/off model can be considered a simple example of a modulated process, which is a combination of two processes. The basic process is a steady stream of packets at constant rate 1/*T*. The basic process is multiplied by a modulating process which is the Markov chain shown in Figure 6 (the active state is equivalent to 1 while the idle state is equivalent to 0). The modulation has the effect of canceling the packets during the idle state. The resulting on/off traffic is shown in Figure 7.

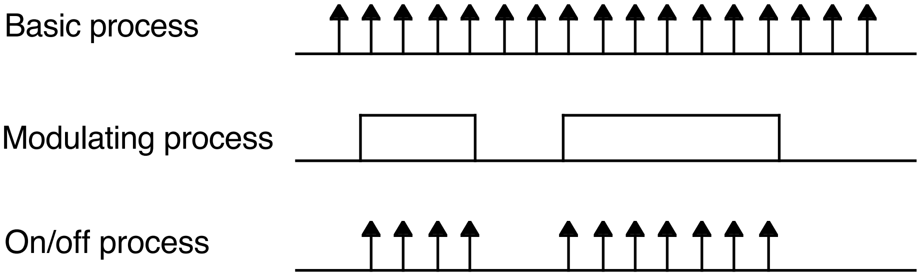

Fig. 7. On/off traffic as example of a modulated process.

Suppose that packets are generated as a Poisson process with rate λ during active periods, instead of a constant rate process. This process is an interrupted Poisson process (IPP). The effect of the modulating process is to cancel or "interrupt" a normal Poisson process during idle states.

Markov modulated Poisson process (MMPP)

While the Poisson process is attractive as a traffic model, it is obviously unrealistic in one aspect: the arrival rate λ is constant. If we consider the aggregate of multiple packet speech flows, the flow rate will not be constant because speech conversations are starting and terminating at random times. The flow rate will drop whenever a speech conversation stops and will increase whenever a new conversation begins.

Suppose *N* speech conversations are multiplexed, and each speech conversation is an IPP. The aggregate flow will be a Markov modulated Poisson process (Heffes and Lucantoni, 1986). The basic process is a Poisson process with rate $\lambda(t)$. The rate is modulated as $\lambda(t) = n(t)\lambda$ where *n(t)* is the number of speech conversations that are active at time *t*. In this case, *n(t)* is the state of a continuous-time Markov chain (specifically a birth-death process) shown in Figure 8. The MMPP preserves some of the memoryless property of the Poisson process and can be analyzed by Markov theory. However, the complexity of the analysis increases with *N* so the value of *N* is usually small.

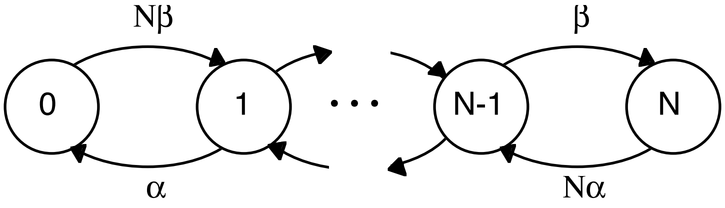

Fig. 8. Birth-death process for number of active sources.

Stochastic fluid model

Fluid models depart from traditional point process models by ignoring the discrete nature of packets. Consider again the MMPP model. The traffic rate is $X(t) = n(t)\lambda$ where $n(t)$ is again the number of active sources at time *t* represented by the birth-death process in Figure 8. The packet process is a complicated point process if one attempts to characterize the packet arrival times $\{t_1, t_2, \ldots\}$ shown in Figure 9.

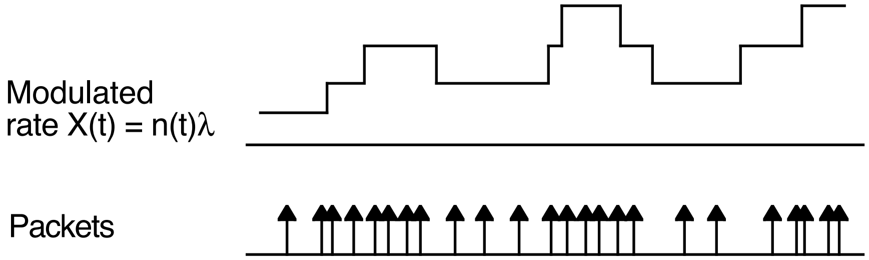

Fig. 9. Packet arrivals in MMPP model.

Instead, it is easier to simply use the process *X(t)* as the traffic model for most purposes (Cassandras, Wardi, Melamed, Sun, and Panayiotou, 2002). According to the stochastic fluid model, an active source transmits data at a constant rate λ like a fluid flow instead of discrete packets. The stochastic fluid model is the aggregation of *N* such fluid sources. The stochastic fluid buffer is attractive for its simplicity and tractability for fluid buffer flow analysis (Anick, et al., 1982).

Fractional Brownian motion

The definition of fractional Brownian motion (FBM) is similar to the well known Brownian motion except generalized to depend on a Hurst parameter *H.* A fractional Brownian motion *X(t)* is a Gaussian continuous process starting at $X(0) = 0$, with zero mean and variance $\text{var}(X(t)) = t^{2H}$, and has stationary increments that are normally distributed with zero mean and variance $\sigma^2 t^{2H}$. Its autocorrelation function is

$$R(s,t) = E(X(s)X(t)) = \tfrac{1}{2}(s^{2H} + t^{2H} - |s-t|^{2H}) \qquad (8)$$

which shows that regular Brownian motion is a special case when $H = 0.5$. For $0.5 < H < 1$, the autocorrelation function exhibits long range dependence with Hurst parameter *H*.

Fractional Brownian motion has been proposed as a model for “free traffic” (unconstrained by network resources) aggregated from many independent sources (Norros, 1995). The model is attractive because it is fairly straightforward for analysis and characterization of long range dependence. It is also tractable for queueing (fluid buffer) analysis, unlike most discrete-time models.

DISCRETE-TIME SOURCE MODELS

In discrete time, we are interested in a stochastic process X_n to represent the source rate sampled at discrete times *n* = 1, 2,… In a sense, there is little difference between continuous-time and discrete-time models because there are stochastic processes in continuous or discrete time that behave similarly to each other. Discrete-time processes can be closely approximated by continuous-time processes, and vice versa. Discrete-time models may be preferred when analysis is easier in discrete time, or when network simulations are carried out in discrete time (versus discrete event simulations).

Time series are commonly assumed to be consecutive measurements of some random phenomenon taken at equally spaced time intervals. Classical time series analysis using ARIMA (autoregressive integrated moving average) models largely follows the Box and Jenkins (1970) methodology.

Time series

The simplest time series model is perhaps the autoregressive (AR) process. A *p*-th order autoregressive model, denoted AR(*p*), has the form

$$X_n = (\alpha_1 X_{n-1} + \cdots + \alpha_p X_{n-p}) + e_n \qquad (9)$$

where $\{\alpha_1, \ldots, \alpha_p\}$ are coefficients and e_n is the residual process assumed to be i.i.d. zero-mean normal random variables. Each point in the process is a linear combination of previous points and a random noise. The coefficients are found by linear regression, or fitting the exponential autocorrelation function to the observed autocorrelation function.

Autoregressive (AR) models are particularly motivated by studies of packet video with low motion (e.g., videoconferencing) where X_n is the bit rate of the *n*-th video frame (Maglaris, et al., 1988). There is strong correlation between the bit rates of successive frames due to the nature of low-motion video and interframe coding algorithms.

An autoregressive moving average (ARMA) model of order (*p,q*), denoted ARMA(*p,q*), has a more general form than the AR process:

$$X_n = (\alpha_1 X_{n-1} + \cdots + \alpha_p X_{n-p}) + (e_n + \beta_1 e_{n-1} + \cdots + \beta_q e_{n-q}) \qquad (10)$$

where $\{\alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q\}$ are coefficients. The moving average terms constitute a linear combination of prior noise samples. The ARMA process has also been proposed for a video source model (Grunenfelder, 1991).

The autoregressive integrated moving average process, denoted ARIMA(*p,d,q*) with integer *d*, is a generalization of ARMA processes. Define *B* as the backward operator $BX_n = X_{n-1}$. A succinct representation of an ARMA process is

$$\phi(B) X_n = \theta(B) e_n \qquad (11)$$

where the polynomials are $\phi(B) = 1 - \alpha_1 B - \cdots - \alpha_p B^p$ and $\theta(B) = 1 + \beta_1 B + \cdots + \beta_q B^q$. In addition, define the difference operator $\nabla X_n = X_n - X_{n-1}$. An ARIMA(*p,d,q*) process has the form:

$$\phi(B)\nabla^d X_n = \theta(B)e_n \tag{12}$$

The parameter *d* is the number of times that the time series is differenced (i.e., converting each *i*th element of the series into its difference from the (*i-j*)th element). The reason for differencing is usually to convert a nonstationary time series into a stationary series by removing seasonal dependencies. The parameters of the autoregressive and moving average components are computed for the series after the differencing is done.

A fractional ARIMA, denoted F-ARIMA(*p,d,q*), is a generalization of an ARIMA allowing non-integer values of *d*. With $0 < d < 0.5$, the F-ARIMA(*p,d,q*) is stationary with long range dependence with Hurst parameter $H = d + 0.5$.

Box-Jenkins methodology

The Box-Jenkins method proceeds in three stages: model identification; model estimation; and model validation. Model identification is perhaps the most challenging and depends largely on the expert judgment of the analyst. The analyst needs to examine the stationarity of the time series, including seasonality (periodic components) and trends (non-periodic linear or non-linear components). Seasonality and trends are subtracted from the series in order to obtain a stationary process. Differencing is another technique to convert a non-stationary series into a stationary one. After stationarity is achieved, the analyst must decide on the order (*p,q*) of the ARMA model, largely based on examination of the autocorrelation function.

With the order of the ARMA process decided, the next step is estimation of the model parameters (coefficients). The most common methods are least squares regression or maximum likelihood estimation.

The last step examines the validity of the chosen ARMA model. If the model was fit properly, the residuals should appear like white noise. That is, residuals should be stationary, mutually independent, and normally distributed.

APPLICATION-SPECIFIC MODELS

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

It is possible to apply the general-purpose continuous-time and discrete-time traffic models from the previous section. However, it is logical that specialized traffic models may work better for particular applications. Therefore, traffic analysts have attempted to search for models tailored to the most common Internet applications (Web, peer-to-peer, and video).

Web traffic

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

An early study focused on the measurement of self-similarity in Web traffic (Crovella and Bestavros, 1997). Self similarity had already been found in Ethernet traffic. Based on days of traffic traces from hundreds of modified Mosaic Web browser clients, it was found that Web traffic exhibited self-similarity with the estimated Hurst parameter H in the range 0.7-0.8. More interestingly, it was theorized that Web clients could be characterized as on/off sources with heavy-tailed on periods. In previous studies, it had been demonstrated that self-similar traffic can arise from the aggregation of many on/off flows that have heavy-tailed on or off periods. A probability distribution is heavy tailed if the asymptotic shape of the distribution is hyperbolic:

$$\Pr(X > x) \sim x^{-\gamma} \qquad (13)$$

as $x \to \infty$, for $0 < \gamma < 2$. Heavy tails implies that very large values have non-negligible probability. For Web clients, on periods represent active transmissions of Web files, which were found to be heavy tailed. The off periods were either "active off", when a user is inspecting the Web browser but not transmitting data, or "inactive off", when the user is not using the Web browser. Active off periods could be fit with a Weibull probability distribution, while inactive off periods could be fit by a Pareto probability distribution. The Pareto distribution is heavy tailed. However, the self-similarity in Web traffic was attributed mainly to the heavy-tailed on periods (or essentially, the heavy-tailed distribution of Web files).

A study of 1900 Web browser clients also characterized Web traffic with an on/off model (Choi and Limb, 1999). Off periods include the user viewing the browser or doing something else. Off periods were fit with a Weibull probability distribution, consistent with earlier studies. The on period again represents a retrieved Web page, but the model was refined with a more detailed view of a Web page as shown in Figure 10. A Web page consists of a main object, which is an HTML document, and multiple in-line objects that are linked to the main object. Both main object sizes and in-line object sizes were fit with (different) lognormal probability distributions. The number of in-line objects associated with the same main object was found to follow a gamma probability distribution. In the same Web page, the intervals between the start times of consecutive in-line objects was fit with a gamma probability distribution.

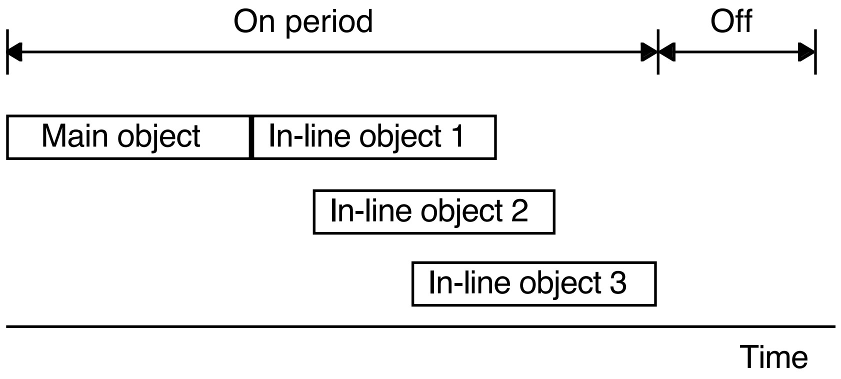

Fig. 10. On/off model with Web page consisting of a main object linked to multiple in-line objects.

It has been argued that TCP flow models are more natural for Web traffic than "page-based" models (Cao, Cleveland, Gao, Jeffay, Smith, and Weigle, 2004). Requests for TCP connections originate from a set of Web clients and go to one of multiple Web servers. A model parameter is the random time between new TCP connection requests. A TCP connection is held between a client and server for the duration of a random number of request/response transactions. Additional model parameters include: sizes of request messages; server response delay; sizes of response messages; and delay until the next request message. Statistics for these model parameters were collected from days of traffic traces on two high-speed Internet access links.

Another study examined 24 hours of flow records from NetFlow-capable routers in the Abilene backbone network (Meiss, Menczer, and Vespignani, 2005). Web traffic composed 42 percent (319 million flows) of the total observed traffic. The Web traffic flows were mapped to a graph of 18 million nodes (Web clients and servers) and 68 million links (flows between client-server pairs). The weight assigned to a link reflects the aggregate amount of data between the nodes. The traffic study examined the probability distribution for number of servers contacted by a client; number of clients handled by a server; amount of data sent by a client; and amount of downloads handled by a server. One of the observations is that both client and server behaviors vary extremely widely, and therefore a "typical" client or server behavior is hard to characterize.

Peer-to-peer traffic

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

Since P2P applications have emerged only in the last few years, there are few traffic studies to date. They have understandably focused more on collecting statistics than the harder problem of stochastic modeling. A recent study examined the statistics of traffic on a university campus network (Saroiu, Gummadi, Dunn, Gribble, and Levy, 2002). The observed P2P applications were Kazaa and Gnutella. The bulk of P2P traffic was AVI and MPEG video, and a small portion was MP3 audio. With this mix of traffic, the average P2P file was about 4 MB, three orders of magnitude larger than the average Web document. A significant portion of P2P was dominated by a few very large objects (around 700 MB) accessed 10-20 times. In other words, a small number of P2P hosts are responsible for consuming a large portion of the total network bandwidth.

Another study collected P2P traffic traces from NetFlow-capable routers in an ISP backbone network (Sen and Wang, 2004). The observed P2P applications were Gnutella, Kazaa, Grokster, Morpheus, and DirectConnect. Again, an extreme skew in the traffic was observed; a small number of P2P hosts (called heavy hitters) accounted for much of the total traffic. The flow interarrival time, defined as the time between the host finishing one flow and then starting another flow, was found to be clustered at 1, 60, and 500 seconds. The very short (one second) interarrival time could be the result of programmed sequential downloading of multiple files. The reason for the 60 and 500 second interarrival times was not clear.

Video

The literature on video traffic modeling is vast because the problem is very challenging. The characteristics of compressed video can vary greatly depending on: the type of video which determines the amount of motion (from low motion videoconferencing to high motion broadcast video) and frequency of scene changes; and the specific intraframe and interframe video coding algorithms. Generally, interframe encoding takes the differences between consecutive frames, compensating for the estimated motion of objects in the scene. Hence, more motion means that the video traffic rate is more dynamic, and more frequent scene changes is manifested by more "spikes" in the traffic rate.

While many video coding algorithms exist, the greatest interest has been in the MPEG (Moving Picture Coding Experts Group) standards. Approved in 1992, MPEG-1 video coding was designed for reasonable quality video at low bit rates. MPEG-2 video encoding at higher bit rates (such as DVDs) was approved in 1994. MPEG-4 approved in 1998 is highly efficient video coding for broadcast and interactive environments. In 2003, a more efficient coding method was approved jointly by the ITU-T as H.264 and MPEG as MPEG-4 Part 10 (also known as Advanced Video Coding).

Video source rates are typically measured frame by frame (which are usually 1/30 sec apart). That is, X_n represents the bit rate of the *n*-th video frame. Video source models address at least these statistical characteristics: the probability distribution for the amount of data per frame; the correlation structure between frames; times between scene changes; and the probability distribution for the length of video files.

Due to the wide range of video characteristics and available video coding techniques, it has seemed impossible to find a single universal video traffic model (Heyman and Lakshman, 1996). The best model varies from video to video file. A summary of research findings:

- The amount of data per frame have been fit to lognormal, gamma, and Pareto probability distributions or combinations of them.
- The autocorrelation function for low motion video is roughly exponential (Maglaris, Anastassiou, Sen, Karlsson, and Robbins, 1988). The autocorrelation function for broadcast video is more complicated, exhibiting both short range dependence at short lags and long range dependence at much longer lags (Ma and Ji, 2001). Time series,

Markov modulated, and wavelet models are popular to capture the autocorrelation structure (Dai and Loguinov, 2005).

- The lengths of scenes usually follow Pareto, Weibull, or gamma probability distributions.
- The amount of data in scene change frames are uncorrelated and follow Weibull, Gamma, or unknown probability distributions.
- The lengths of streaming video found on the Web appear to have a long-tailed Pareto probability distribution (Li, Claypool, Kinicki, and Nichols, 2005).

ACCESS REGULATED SOURCES

Traffic sources can not typically transmit data without any constraints. To be more realistic, traffic models must taking the limiting factors into account. For one thing, networks can regulate or “police” the rate of source traffic at the user-network interface in order to prevent congestion. Access policing is more common in networks where connections are established by a signaling protocol. Signaling messages inform the network about traffic characteristics (e.g., peak rate, mean rate) and request a QoS. If a connection is accepted, the QoS is guaranteed as long as source traffic conforms to the given traffic parameters. A policer enforces conformance to given traffic parameters by limiting the source traffic as necessary.

Leaky bucket regulated sources

The simplest policer enforces a peak rate defined by the shortest allowable interarrival time between two consecutive packets. If packets are arriving too closely in time, the second packet is nonconforming and therefore discarded. However, a policer with more tolerance for minor deviations from the peak rate is preferred.

The leaky bucket algorithm is typically used for access policing because its enforcement is flexible dependent on adjustment of two parameters: the bucket size *B* and a leak rate *R*. Packets are conforming and admitted into the network if they can increment the bucket contents by their payload size without overflowing. The contents of the bucket are emptied at the leak rate *R*. The leak rate *R* is the long-term average rate allowed. A larger bucket size allows greater variations from the rate *R*. A burst at a rate higher than *R* can still be conforming as long as the burst is not long enough to overflow the bucket. Hence the maximum burst length *b* is directly related to the bucket size.

The maximum amount of traffic allowed by a leaky bucket policer in an interval (0,t) is represented by the deterministic (σ, ρ) calculus (Cruz, 1991). The allowable traffic is deterministically bounded by

$$\int_0^t X(s)ds \leq \sigma + \rho t \tag{14}$$

where $\sigma > 0$, $\rho > 0$. From (14), it is clear that ρ is the long-term average rate, while σ is an allowable burst size.

Bounding interval-length dependent (BIND) model

The deterministic bounding interval-length dependent (D-BIND) model characterizes traffic sources by multiple rate-interval pairs (R_k, I_k). R_k is the worst case rate measured over every interval of length I_k. For practical applications, only a limited number of pairs would be used to characterize a source (Knightly and Zhang, 1997).

S-BIND is a stochastic version where a traffic source is similarly characterized by multiple pairs. In this case, R_k is a random variable that is stochastically larger than the rate over any interval of length I_k. Suppose r_k is the traffic rate in interval I_k, then $\Pr(R_k > x) \geq \Pr(r_k > x)$.

CONGESTION-DEPENDENT FLOWS

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

TCP flows with congestion avoidance

TCP senders follow a self-limiting congestion avoidance algorithm that adjusts the size of a congestion window according to the inferred level of congestion in the network. The congestion window is adjusted by so-called additive increase multiplicative decrease (AIMD). In the absence of congestion when packets are acknowledged before their retransmission time-out, the congestion window is allowed to expand linearly. If a retransmission timer expires, TCP assumes that the packet was lost and infers that the network is entering congestion. The congestion window is collapsed by a factor of a half. Thus, TCP flows generally have the classic "sawtooth" shape (Firoiu, et al., 2002; Paxson, 1994).

Accurate models of TCP dynamics are important because of TCP's dominant effect on Internet performance and stability. Hence, many studies have attempted to model TCP dynamics (Mathis, et al., 1997; Padhye, 2000; Barakat, 2001; Sikdar, Kalyanaraman, and Vastola, 2003;

Altman, Avrachenkov, and Barakat, 2005). Simple equations for the average throughput exist, but the problem of characterizing the time-varying flow dynamics is generally difficult due to interdependence on many parameters such as the TCP connection path length, round trip time, probability of packet loss, link bandwidths, buffer sizes, packet sizes, and number of interacting TCP connections.

TCP flows with active queue management

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

Active queue management schemes, mainly random early detection (RED) and its many variants, eliminate the synchronization phenomenon and thereby try to achieve better utilization and smaller queue lengths (Floyd and Jacobson, 1993; Firoiu and Borden, 2000). Instead of waiting for the buffer to overflow, RED drops a packet randomly with the goal of losing packets from multiple TCP flows at different times. Thus, TCP flows are unsynchronized, and their aggregate rate is smoother than before. From a queueing theory viewpoint, smooth traffic achieves the best utilization and shortest queue.

Active queue management has stimulated a number of control theory approaches to understanding the stability, fairness issues, and flow dynamics (Tan, Zhang, Peng, and Chen, 2006; Srikant, 2004).

CONCLUSIONS

Proper network design requires an understanding of the source traffic entering the network. Many continuous-time and discrete-time traffic models have been developed based on traffic measurement data. The choice of traffic models involves at least two major considerations. The first consideration is accuracy. Accurate traffic models should include not only source behavior but also possible policing or congestion avoidance. The second consideration is ease of queueing analysis. Traffic models are useful only if network performance can be evaluated. It is always possible to evaluate network performance via computer simulations, but analysis is preferred whenever analysis is tractable.

GLOSSARY

Access policing: regulation of ingress traffic at the user-network interface.

Active queue management (AQM): intelligent buffer management strategies, mainly random early detection (RED) and its variants, enabling traffic sources to respond to congestion before buffers overflow.

Autoregressive moving average (ARMA): a general type of time series model including linear dependence on previous data samples and linear dependence on noise samples.

Burstiness: a measure of the variability of a traffic source rate.

Fluid model: a view of data traffic ignoring the discrete nature of packets.

Fractional Brownian motion (FBM): a long range dependent Gaussian process with stationary increments including Brownian motion as a special case.

Leaky bucket: a flexible algorithm for policing a specific traffic rate with an adjustable tolerance for deviation from that rate.

Long range dependence (LRD): a property of stochastic processes with autocorrelation functions that decay slower than exponentially.

Markov modulated Poisson process (MMPP): a Poisson process where the arrival rate varies according to a continuous-time Markov chain.

On/off model: a stochastic process with alternating active and idle states.

Poisson process: a point process with interarrival times that are independent and identically distributed exponential random variables.

Quality of service (QoS): the end-to-end network performance defined from the perspective of a specific user’s connection.

Self similarity: a property of stochastic processes that preserve statistical characteristics over different timescales.

Time series: a discrete-time stochastic process usually considered to represent measurements of a random phenomenon sampled at regularly spaced times.

Transmission control protocol (TCP): a transport layer protocol used over IP to ensure reliable data delivery between hosts.

REFERENCES

Abry, P., and Veitch, D. (1998). Wavelet analysis of long-range-dependent traffic. IEEE Transactions on Information Theory, 44, 2-15.

Abry, P., Baraniuk, R., Flandrin, P., Riedi, R., and Veitch, D. (2002). Multiscale nature of network traffic. IEEE Signal Processing Magazine, 19, 28-46.

Adas, A. (1997). Traffic models in broadband networks. IEEE Communications Magazine, 35, 82-89.

Anick, D., Mitra, D., and Sondhi, M. (1982). Stochastic theory of a data-handling system with multiple sources. Bell System Technical Journal, 61, 1971-1894.

Barakat, C., Thiran, P., Iannaccone, G., Diot, C., and Owezarski, P. (2003). Modeling Internet backbone traffic at the flow level. IEEE Transactions on Signal Processing, 51, 2111-2124.

Beran, J. (1994). Statistics for Long-Memory Processes. London: Chapman and Hall.

Box, G., and Jenkins, G. (1970). Time-Series Analysis, Forecasting and Control. San Francisco, CA: Holden-Day.

Brownlee, N., and Claffy, K. (2002). Understanding Internet traffic streams: dragonflies and tortoises. IEEE Communications Magazine, vol. 40, 110-117.

Cao, J., Cleveland, W., Gao, Y., Jeffay, K., Smith, F., and Weigle, M. (2004). Stochastic models for generating synthetic HTTP source traffic. In proc. of IEEE Infocom 2004, 1546-1557.

Cassandras, C., Wardi, Y., Melamed, B., Sun, G., and Panayiotou, C. (2002). Perturbation analysis for online control and optimization of stochastic fluid models. IEEE Transactions on Automatic Control, 47, 1234-1248.

Choi, H-K., and Limb, J. (1999). A behavioral model of Web traffic. In proc. of 7th International Conf. on Network Protocols (ICNP'99), 327-334.

Crovella, M., and Bestavros, A. (1997). Self-similarity in World Wide Web traffic: evidence and possible causes. IEEE/ACM Transactions on Networking, 5, 835-846.

Cruz, R. (1991). A calculus for network delay, part I: network elements in isolation. IEEE Transactions on Information Theory, 37, 114-131.

Dai, M., and Loguinov, D. (2005). Analysis and modeling of MPEG-4 and H.264 multi-layer video traffic. In proc. of IEEE Infocom 2005, 2257-2267.

Erramilli, A., Roughan, M., Veitch, D., and Willinger, W. (2002). Self-similar traffic and network dynamics. Proceedings of the IEEE, 90, 800-819.

Fendick, K., Saksena, V., and Whitt, W. (1991). Investigating dependence in packet queues with the index of dispersion for work. IEEE Transactions on Communications, 39, 1231-1244.

Firoiu, V., and Borden, M., (2000). A study of active queue management for congestion control. In proc. of IEEE Infocom 2000, 1435-1444.

Firoiu, V., Le Boudec, J.-Y., Towsley, D., and Zhang, Z.-L. (2002). Theories and models for Internet quality of service. Proceedings of the IEEE, 90, 1565-1591.

Floyd, S., and Jacobson, V. (1993). Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking, 1, 397-413.

Fraleigh, C., Moon, S., Lyles, B., Cotton, C., Khan, M., Moll, D., Rockell, R., Seely, T., and Diot, C. (2003). Packet-level traffic measurements from the Sprint IP backbone. IEEE Network, 17, 6-16.

Frost, V., and Melamed, B. (1994). Traffic modeling for telecommunications networks. IEEE Communications Magazine, 32, 70-81.

Gilbert, A., Willinger, W., and Feldman, A. (1999). Scaling analysis of conservative cascades, with applications to network traffic. IEEE Transactions on Information Theory, 45, 971-991.

Grunenfelder, R., Cosmas, J., Manthorpe, S., and Odinma-Okafor, A. (1991). Characterization of video codecs as autoregressive moving average processes and related queueing system performance. IEEE Journal on Selected Areas in Communications, 9, 284-293.

Gusella, R. (1991). Characterizing the variability of arrival processes with indexes of dispersion. IEEE Journal on Selected Areas in Communications, 9, 203-211.

Heffes, H., and Lucantoni, D. (1986). A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. IEEE Journal on Selected Areas in Communications, SAC-4, 856-868.

Heyman, D., and Lakshman, T. (1996). Source models for VBR broadcast-video traffic. IEEE/ACM Transactions on Networking, 4, 40-48.

Kelly, F. (1996). Notes on effective bandwidths. In Stochastic Networks: Theory and Applications (pp. 141–168). Oxford: Oxford University Press.

Kleinrock, L. (1976). Queueing Systems Volume I. NY: Wiley & Sons.

Knightly, E., and Zhang, H. (1997). D-BIND: an accurate traffic model for providing QoS guarantees to VBR traffic. IEEE/ACM Transactions on Networking, 5, 219-231.

Leland, W., Taqqu, M., Willinger, W., and Wilson, D. (1994). On the self-similar nature of Ethernet traffic (extended version). IEEE/ACM Transactions on Networking, 2, 1-15.

Li, M., Claypool, M., Kinicki, R., and Nichols, J. (2005). Characteristics of streaming media stored on the Web. ACM Transactions on Internet Technology, 5, 601-626.

Ma, S., and Ji, L. (2001). Modeling heterogeneous network traffic in wavelet domain. IEEE/ACM Transactions on Networking, 9, 634-649.

Maglaris, B., Anastassiou, D., Sen, P., Karlsson, G., and Robbins, J. (1988). Performance models of statistical multiplexing in packet video communications. IEEE Transactions on Communications, 36, 834-844.

Mathis, M., Semke, J., Mahdavi, J., and Ott, T. (1997). The macroscopic behavior of the TCP congestion avoidance algorithm. Computer Communications Review, 27, 67-82.

Meiss, M., Menczer, F., and Vespignani, A. (2005). On the lack of typical behavior in the global Web traffic network. In proc. of 14th International World Wide Web Conf., 510-518.

Michiel, H., and Laevens, K. (1997). Teletraffic engineering in a broadband era. Proceedings of the IEEE, 85, 2007-2033.

Norros, I. (1995). On the use of fractional Brownian motion in the theory of connectionless networks. IEEE Journal on Selected Areas in Communications, 13, 953-962.

Padhye, J., Firoiu, V., Towsley, D., and Kurose, J. (2000). Modeling TCP Reno performance: a simple model and its empirical validation. IEEE/ACM Transactions on Networking, 8, 133-145.

Park, K., and Willinger, W., eds. (2000). Self-Similar Network Traffic and Performance Evaluation. NY: Wiley & Sons.

Paxson, V. (1994). Empirically derived analytic models of wide-area TCP connections. IEEE/ACM Transactions on Networking, 2, 316-336.

Paxson, V., and Floyd, S. (1995). Wide area traffic: the failure of Poisson modeling. IEEE/ACM Transactions on Networking, 3, 226-244.

Riedi, R., Crouse, M., Ribeiro, V., and Baraniuk, R. (1999). A multifractal wavelet model with application to network traffic. IEEE Transactions on Information Theory, 45, 992-1018.

Roberts, J. (2004). Internet traffic, QoS, and pricing. Proceedings of the IEEE, 92, 1389-1399.

Roughan, M., Veitch, D., and Abry, P. (2000). Real-time estimation of the parameters of long-range dependence. IEEE/ACM Transactions on Networking, 8, 467-478.

Sahinoglu, Z, and Tekinay, S. (1999). On multimedia networks: self-similar traffic and network performance. IEEE Communications Magazine, 37, 48-52.

Saroiu, S., Gummadi, K., Dunn, R., Gribble, S., and Levy, H. (2002). An analysis of Internet content delivery systems. ACM SIGOPS Operating Systems Review, 36, 315-327.

Sen, S., and Wang, J. (2004). Analyzing peer-to-peer traffic across large networks. IEEE/ACM Transactions on Networking, 12, 219-232.

Shim, C., Ryoo, I., Lee, J., and Lee, S. (1994). Modeling and call admission control algorithm of variable bit rate video in ATM networks. IEEE Journal on Selected Areas in Communications, 12, 332-344.

Srikant, R. (2004). The Mathematics of Internet Congestion Control. Boston: Birkhauser.

Sriram, K., and Whitt, W. (1986). Characterizing superposition arrival processes in packet multiplexers for voice and data. IEEE Journal on Selected Areas in Communications, SAC-4, 833-846.

Tan, L., Zhang, W., Peng, G., and Chen, G. (2006). Stability of TCP/RED systems in AQM routers, IEEE Transactions on Automatic Control, 51, 1393-1398.

Thajchayapong, S., and Peha, J. (2006). Mobility patterns in microcellular wireless networks. IEEE Transactions on Mobile Computing, 5, 52-63.

Thompson, K., Miller, G., and Wilder, R. (1997). Wide-area Internet traffic patterns and characteristics. IEEE Network, 11, 10-23.

Willinger, W., and Paxon, V. (1998). Where mathematics meets the Internet. Notices of the American Mathematical Society, 45, 961-970.

Wu, C-H., Lin, H-P., and Lan, L-S. (2002). A new analytic framework for dynamic mobility management of PCS networks. IEEE Transactions on Mobile Computing, 1, 208-220.

FURTHER READING

The literature on traffic models is quite extensive, including references where traffic models are discussed in combination with performance analysis. Examples of useful books covering both traffic modeling and performance analysis include:

Jain, R. (1991). The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. NY: Wiley & Sons.

Krishna, M., Gadre, V., and Desai, U. (2003). Multifractal based Network Traffic Modeling. Boston, MA: Kluwer Academic Publishers.

Kumar, A., Manjunath, D., and Kuri, J. (2004). Communication Networking: An Analytical Approach. San Francisco, CA: Morgan Kaufmann Publishers.

A variety of traffic traces are publicly available on the Web. These can easily be found by searching. Popular collections include:

Internet Traffic Archive, available at http://ita.ee.lbl.gov/html/traces.html, date of access: Sept. 19, 2006.

NLANR network traffic packet header traces, available at http://pma.nlanr.net/Traces/, date of access: Sept. 19, 2006.

MPEG-4 and H.263 video traces, available at http://www-tkn.ee.tu-berlin.de/research/trace/trace.html., date of access: Sept. 19, 2006.