Studies on the Impact of Long-Term Correlation on Computer Network Performance

Hany D. Alsaialy

CENG 98-30

Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, California 90089-2562
(213-740-4579)
December 1998

STUDIES ON THE IMPACT OF LONG-TERM CORRELATION ON COMPUTER NETWORK PERFORMANCE

by

Hany D. Alsaialy

A Thesis Presented to the
FACULTY OF THE SCHOOL OF ENGINEERING
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
Master of Science
(Computer Engineering)

December 1998

Contents

List of Figures

List of Tables

Abstract

STUDIES ON THE IMPACT OF LONG-TERM CORRELATION ON COMPUTER NETWORK PERFORMANCE

Hany D. Alsaialy

In Part I of this thesis we present a framework for modeling Variable-Bit-Rate (VBR) traffic based on semi-Markov processes. We propose an algorithm based on a simple clustering method to build semi-Markov models for computer network traffic modeling. We introduce our novel mechanism by giving its detailed algorithm, and later analyze its performance by means of simulation. We reveal the efficacy of the proposed method for Long-Range Dependence (LRD) traffic modeling under realistic buffer sizes, and compare the performance of a real computer network VBR traffic trace with the synthesized traces obtained using our mechanism.

In Part II, we present a mechanism for the modeling of the Internet's World Wide Web (WWW or Web) traffic based on the superposition of independent and identically distributed (i.i.d.) point processes. We describe the Superposition of Fractal Renewal Processes (Sup-FRP) traffic model used for modeling Web request arrivals. We show that neglecting the correlation found in real Web requests will lead to inaccurate performance evaluations. We show the performance impact of correlated Web requests on throughput and average response time, and compare our findings with previously reported results.

Acknowledgments

I am indebted to a large number of people who contributed directly and indirectly in the development of this thesis.

First and foremost, I would like to thank Professor John Silvester, my academic advisor, whose constant supervision and perceptive insights helped me finish this work. I am also very thankful to Dr. Bo Ryu from the Hughes Research Labs (HRL) without whom, part II of this thesis wouldn't have been possible. I would like to thank Professor Deborah Estrin and Dr. Antonio Ortega for serving on my committee.

I would like to acknowledge my sponsoring agency, the Institute of Public Administration (IPA), Riyadh, Saudi Arabia, for their support throughout my studying in the United States. I would also like to acknowledge the support of the HRL Labs, Malibu, California, USA, for their partial support of part II of this thesis.

Last, but by no means least, I would like to thank my Mom and Dad. For their unconditional Love and constant support. To my brother Sami, and everyone who contributed and supported me throughout this work. I thank You all.

HANY D. ALSAIALY

Part I

Link-layer Modeling

Chapter 1

Introduction

In recent years, there has been significant attention paid to computer network traffic modeling. This research has increased tremendously since the discovery of Long-Range Dependence (LRD) in real computer network traffic streams [LTWW94]. In [LTWW94] several traces of real computer traffic were collected over a three year period from both Local Area Network (LAN) and Wide Area Network (WAN). When they were analyzed they revealed what is now commonly known as LRD. But what does LRD mean? Simply, and with a high level of abstraction, LRD states that "things that happen far apart cannot be considered independent". Later, we will give several mathematical characterizations of LRD. After the first results published revealing the omnipresence of LRD in computer network traffic streams, other research studies where done and similar results obtained [GW94]. An example of such a stream is the one from the Star Wars movie encoded using the MPEG standard.

When modeling computer network traffic, it is important to identify the various types of traffic that are found in today's networks and the quality of service (QOS) expected by the user.

It is clear that the difference in network traffic modeling lies not only on the type of data being transferred (e.g. voice, video, data) or whether the data being

transferred is encoded or not, but also on the QOS required by the user. As noted, there is a relationship between the QOS (that concerns the user), and resources that need to be allocated in order to guarantee the service required (that concerns network management). Unfortunately, there is an inverse relation between QOS and network utilization (e.g. resource allocation), in other words, as utilization increases, overall delays increase which adversely affects the QOS of the users. Both Short-Range Dependence (SRD) and LRD phenomena may impact this utilization/QOS trade-off, so it is important to use traffic models that appropriately represent both factors.

Before proceeding, we emphasize two different aspects of network performance: i) network management, and ii) user requirements. We will later reveal their importance to the concept of traffic modeling. We identify three fundamental constraints [CG96], each constraint affects overall network utilization, or the user performance (i.e. QOS). These three main constraints are:

1. (Maximal) Delay constraint.

2. (Maximal) Loss constraint.

3. (Minimal) Utilization constraint.

Figure 1.1 shows the constraints graphically[1]. It is important to keep these constraints in mind when designing computer network traffic models.

1.1 Long-Range Dependence in Traffic streams

In the introduction we gave a brief description of LRD, we now present the concept of LRD from a mathematical point of view. There are several ways to

[1]Throughout this thesis "frame" refers to "frame-time"=$\frac{1}{24}$second.

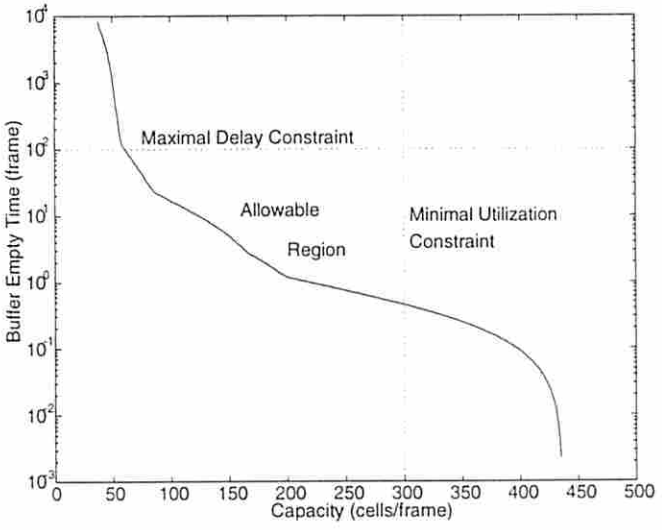

Figure 1.1: The main fundamental constraints in ATM networks [CG96].

describe the concept of LRD, we briefly present some methods, more details can be found in [LTWW94].

Let $X = \{X_i : i \geq 0\}$ be a wide-sense stationary random process. X is LRD if the variance of its aggregate process $X_i^{(m)}$decreases slowly. This can be checked by the variance-time plot. The variance of the aggregate process $X_i^{(m)}$, is defined as the variance of the number of observations in the new process $X_i^{(m)}$. $X_i^{(m)}$is obtained by averaging the original process X over non-overlapping blocks of size m; that is, $X_i^{(m)} = \frac{1}{m}(X_{im-m+1} + \ldots + X_{im})$, $i \geq 1$. Mathematically for a process to be LRD we require $var(X_i^{(m)}) \sim C_1 m^{-D}$ as $m \longrightarrow \infty$, $0 < D < 1$. where $D = 2 - 2H$. H is known as the *Hurst* parameter. For a process to be LRD we require $1/2 < H < 1$. Since we have $D = 0$ for $H = 1$, and $D = 1$ for $H = 1/2$, the presence of LRD can be determined by the slope of the variance-time plot; a slope of 0 (i.e. $H = 1$) will indicate strong LRD while a slope of -1 (i.e. $H = 1/2$) indicates the absence of LRD, that is, the process is SRD.

Other conditions that can be used to test a random process for LRD include; testing the autocovariance function $\gamma(k)$, the power spectral density $f(\omega)$, or the mean of its rescaled adjusted range $R(n)/S(n)$. Mathematically, we require $\gamma(k) \sim C_2 k^{-D}$ as $m \longrightarrow \infty$, $0 < D < 1$, this can be checked by the empirical autocorrelation function. $f(\omega) \sim C_3 \omega^{-\alpha}$ as $\omega \longrightarrow 0$, $0 < \alpha < 1$, this can be checked by the peridiogram. Finally, $E[R(n)/S(n)] \sim C_4 n^{-H}$ as $n \longrightarrow \infty$, $1/2 < H < 1$, this can be checked by the rescaled adjusted range plot (also called the pox diagram of R/S)[BSTW95]. Note that: $D = 1 - \alpha$. See [BSTW95], [GW94],[LTWW94], [LTWW95] for more details.

In our simulations we will look at the variance-time plot, and other tools we later describe.

Chapter 2

Basic Clustering Method and Traffic Models Proposed

In this chapter we introduce the notion of clustering; we first give a generic clustering algorithm, and later introduce the data we wish to cluster.

2.1 How to cluster data

For the time being we introduce the general clustering algorithm, later we will describe the actual data used in our experiments, and see how the clustering method can be used to build the proposed traffic models.

Clustering of data, is performed in the following way:

Algorithm 1 (*Clustering*)

1 *begin read the first sample* x_1, *set* $C_1 \leftarrow x_1$, $n_1 \leftarrow 1$, *and initialize* $Threshold_1$[1]

2 *do read the next sample* x_i

3 *do for all clusters* j *(with centers* C_j*)*

[1] A description of the method we use to calculate the threshold is given in chapter 3.

4 *if there exist a distance, d_{ij}, from x_i to C_j less than the threshold of cluster j, that is;*

$$d_{ij} = \|x_i - C_j\| < Threshold_j$$

5 *then select cluster j having the smallest d_{ij}*

6 *end for*

7 *if cluster j (steps 4, 5) is found then*

8 *update the center of cluster j by:*

$$C_j \leftarrow \frac{1}{n_j + 1}\{n_j C_j + x_i\}$$

where n_j = total number of samples in cluster j before adding x_i.

9 *update $n_j \leftarrow n_j + 1$*

10 *update $Threshold_j$*

11 *else form a new cluster as performed in 1*

12 *until end of stream (or for a reasonable time enough to characterize traffic)*

13 *end*

In our experiments we used a stream of VBR video traffic, and converted the stream representing bits/frame to 53 byte ATM cells/frame. We left the measurement rate of 24 frames/sec unchanged. Hence, clustering is performed per-frame. The stream of the full-length Star Wars movie used was originally coded using the MPEG standard measured at frame resolution. Since this known stream resembles real VBR traffic, we choose it to verify the efficacy of the proposed model. Note that this stream is bursty and exhibits LRD as will be shown later.

2.2 Clustering method and Markov Modeling

In this section we describe the method we use to build a pure Markovian model based on the clusters obtained by the algorithm just described.

The data stream represents a sequence of random events, we will therefore specify the analogy of the clustering method to a Discrete-Time Markov Process (DTMP). As part of the clustering process described above, we also maintain the cluster transition history by the matrix T which specifies the number of transitions t_{ij} from cluster i to cluster j $\forall ij$.

The analogy with the DTMP is now clear and easy to formalize. The mapping is done as follows:

- The number of states of the DTMP $\longleftrightarrow$ Number of clusters.
- The average input rate generated in state i $\longleftrightarrow$ Final location of the cluster center C_i.
- The transition probabilities P_{ij} between states $i, j \longleftrightarrow \frac{t_{ij}}{\sum_j t_{ij}}$ $\forall ij$.

Throughout this thesis, we will use the terms cluster and state (i.e. state of the Markov or semi-Markov chain) interchangeably.

2.3 Clustering Method and Semi-Markov Modeling

We are now ready to define the steps required to build a Semi-Markov Process (SMP) using the clustering method. We would like to mention that a SMP in not Markovian, in other words, a SMP may or may not be SRD depending on the holding-time in each state, for more details see [Nel95]. A SMP is a generalization

of a Markov chain. In such a process we include a pdf to specify the state holding time in each state. Note that if the state holding time follows the geometric or exponential distribution, we would then have a Markov chain. Any other pdf yields to a SMP. In this study, we use holding times distributed according to a Pareto distribution (other holding time distributions could be as well used).

In order to model the holding time H_j for any state j of the semi-Markov chain, we need a sequence of random events representing these holding times. Let $x_{i,j}$ be the random variable counting the number of $j \longrightarrow j$ transitions in a single visit to state j, and m_j be the number of visits to state j observed during the trace, in other words, define $\left\{X_{m_j,j}\right\} = x_{1,j}, \ldots, x_{m_j,j}$.

To estimate the parameters of a pareto pdf defining the holding-time of state j of our SMP, given $\left\{X_{m_j,j}\right\}$, we use the method of maximum likelihood.

The pareto pdf is defined as follows [AFT98]:

$$f\left(x \mid \alpha_j, k_j\right) = \alpha_j k_j^{\alpha_j} x^{-(\alpha_j+1)} \qquad \alpha_j, k_j > 0,\ x \geqslant k_j \tag{2.1}$$

where α_j, k_j are the pareto parameters and the subscript j refers to cluster j. Hence, the likelihood of α_j is:

$$lik\left(\alpha_j\right) = \prod_{i=1}^{m_j} f\left(x_{i,j} \mid \alpha_j, k_j\right) \tag{2.2}$$

and the log-likelihood of α_j is:

$$l\left(\alpha_j\right) = \sum_{i=1}^{m_j} \log f\left(x_{i,j} \mid \alpha_j, k_j\right) \tag{2.3}$$

solving $\frac{\partial}{\partial \alpha_j} l\left(\alpha_j\right) = 0$ for α_j we get:

$$\widehat{\alpha}_{jMLE} = \frac{m_j}{\sum\limits_{i=1}^{m_j} \log x_{i,j} + m_j \log \widetilde{k}_j} \qquad \forall j \tag{2.4}$$

where $\widetilde{k}_j$ is an estimate of k_j that can be found solving for k_j numerically using Newton's method [PVTF92].

We adjust the transition probabilities P_{ij} to eliminate the self loops by:

$$P_{ij} = \begin{cases} \frac{P_{ij}}{(1-P_{ii})} & \forall i \neq j \\ 0 & \forall i = j \end{cases} \tag{2.5}$$

The SMP described operates in the following fashion:

1. After entering state i, find the holding time H_i that the process spends in state i before moving to the next state j.

2. Select the next state $j \neq i$ where the transition is to be made.

This is the simplest definition for operating a SMP. In our simulations we used this definition, however, we would like to describe another that requires further investigation:

1. After entering state i, select the next state $j \neq i$ where the transition is to be made.

2. Find the holding time $H_{i,j}$ that the process will spend state i before moving to the next state j (where $H_{i,j}$ is defined as the holding time in state i given that the next state is j).

In this definition, the state holding time depends on the next state transition, in other words, several pdfs define the holding time in a given state. Intuitively, we believe this approach would yields better modeling results.

Chapter 3

Simulation Results

We use the algorithms described in chapter 2 to build both DTMPs and SMPs by using the VBR video stream from the Star Wars movie, and analyze our results by simulation. We compare several aspects of the original and model generated streams, including, Cell-Loss Rate (CLR) under different network scenarios. We will also compare the behavior of the streams by the Leaky-Bucket Contour plot (LBC) [LNR94]. For the LBC plot method, there are several regions of interest as described in [CG96] see Figure 3.1. The three main regions are: i) LRD dominant region, ii) SRD dominant region, and iii) Marginal distribution dominant region. Notice that in the clustering method, we produce a processed stream creating a smoothed version of the original. The resulting stream will vary in number of clusters depending on the threshold size of the clusters.

The threshold of a given cluster can be fixed (e.g. Parzen-window method), or made sensitive to the location of the cluster (e.g. k-nearest-neighbor method), for more details see [DH73]. In our experiments we use the latter approach[1] by setting the threshold as: $Threshold_j \leftarrow Min[A + (\frac{B}{100} * C_j), thresh_{\max}]$, where A,

[1] Informally speaking, our approach is an intermediate between the two methods since we adjust the size of the threshold during training. This is known as the "Relaxation method". See [DH73].

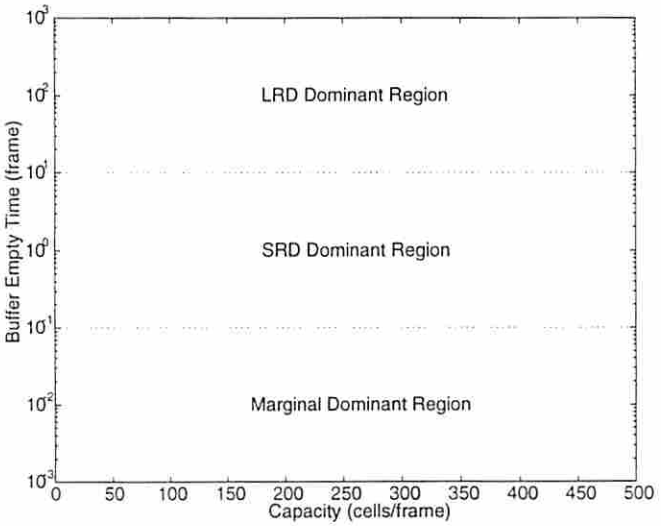

Figure 3.1: Dominant regions for the three main components of a trace [CG95].

B, and $thresh_{\max}$ are set before each simulation run. C_j is the center of cluster j. Since the threshold is set according to the cluster center, we obtained larger thresholds for higher values of C_j; this helped compensate the larger variability for the higher levels. In our experiments we adjusted $A \approx [5, 15]$, $B \approx [10, 20]$, and $thresh_{\max} \approx 90$. From several simulation runs we found that approximately eight clusters gave good results to characterize the original stream. We later show the similarity of the processed stream compared to the original one, and show that the only difference is in the marginal distribution. In other words, both the original and the smoothed stream are similar in terms of LRD and SRD as the LBC plot will verify. The use of eight quantized levels (clusters in our case) to provide an accurate representation of the stream agree with results reported in [SS93].

3.1 Matching Mean, Variance, and CLR

Table 3.1 summarizes some statistical results from our experiments.

Trace	Samples	Mean	Peak/Mean Ratio	Variance	States
MPEG stream	174136	36.289	12.015	1835.554	353
Processed stream	174136	35.914	11.305	1836.113	8
DTMP stream	174136	35.725	11.756	1815.807	7-12
SMP stream	174136	26.124	15.962	1225.562	7-12

Table 3.1: Statistics of the real and synthesized data traces.

The 353 states (or quantized levels) resulted from converting the original stream to ATM cells. In addition, the range in the number of states for the DTMP and SMP stream is due to the fact that several experiments were conducted each with slightly different cluster size. We would also like to mention that we used a deterministic process while in the DTMP/SMP state. We can, however, observe that both a deterministic and a Poisson process would produce similar results by comparing the original and processed stream. We observe that the performance impact (e.g. Cell-Loss Rates CLR) is due to: a) number of states, b) transition probabilities P_{ij}, and c) holding-times H_j. The high-frequency component, on the other hand, has only a marginal impact.

We can see from Table 3.1 that the simple statistics from the DTMP closely matched the average statistics from the original stream. However, we need to investigate the behavior of the streams since we still have no indication of the correlation structure. We further analyze the streams by comparing CLR for different buffer sizes and capacity values, to compare the effects of Long-Range Dependence (LRD) and Short-Range Dependence (SRD). Figures 3.3, 3.4, and 3.5 show the CLR ob-

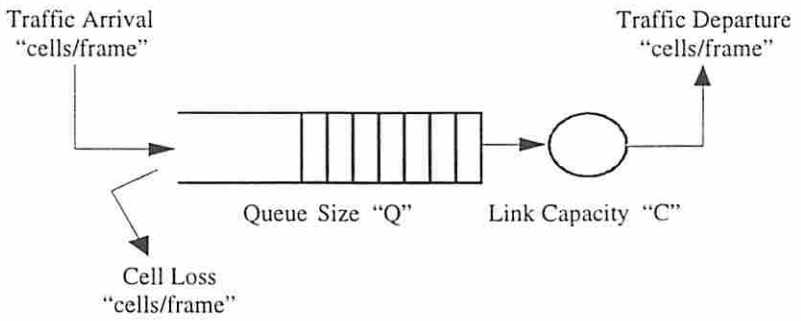

Figure 3.2: A single queue single server system.

tained in our simulation. We applied all four streams to a simple FCFS single queue single server system, see Figure 3.2.

In Figure 3.3 we used a small buffer size for the wide range of capacity allocation (Maximum Queueing Delay (MQD)$\approx$4.5$-$45 ms), all streams seem to match the behavior of the original stream well due to: a) the effect of SRD is pronounced; or b) we see little effect from LRD. Therefore a well designed SRD Markovian model gave good results under these operational conditions (i.e. relatively small buffer).

In Figure 3.4, we fixed the link capacity to 60 cells/frame ($\approx$600Kbps) and showed results over a wide range of buffer sizes (MQD$\approx$7$-$350ms). As the buffer size increases, the effect of LRD become more pronounced making the pure DTMP fail to model the original stream. On the other hand, the SMP model gives better results as expected. We reemphasize the importance of realistic buffer sizes used in simulation due to the QOS expected by the users since as we know, buffer sizes are finite and often small in a real life computer network. More results showing the dominant effect of short-term correlation on CLR can be found in [RE96]. We see, therefore, the efficacy of the proposed semi-Markovian model for VBR traffic modeling.

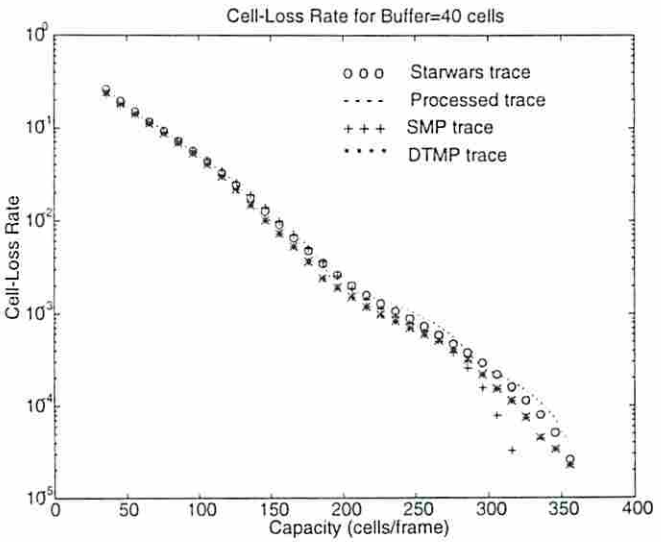

Figure 3.3: Effect of short-range dependence is pronounced with small buffer size.

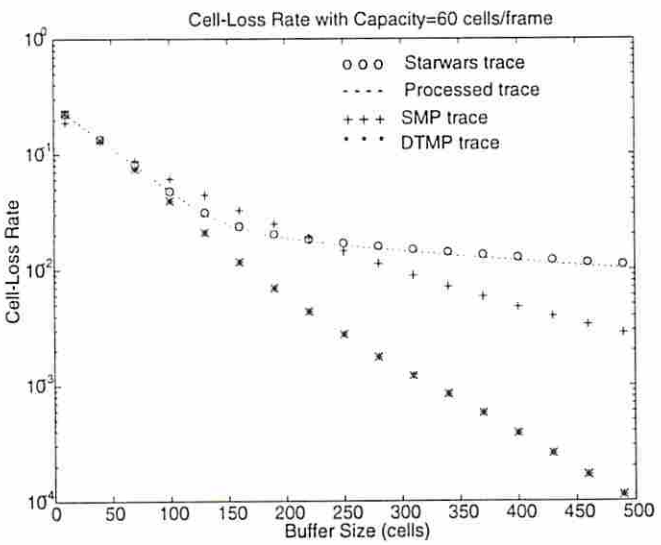

Figure 3.4: Effect of long-range dependence is pronounced as buffer size increase.

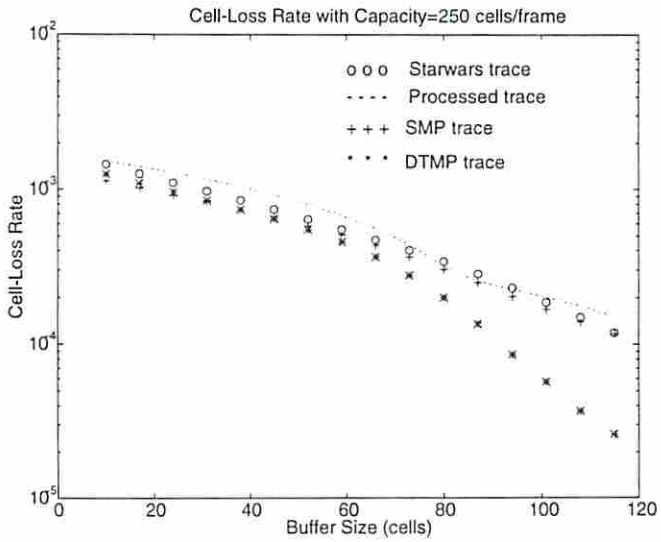

Figure 3.5: The semi-Markov process matches well under realistic operational scenarios.

To validate our results we conducted additional experiments for realistic values of buffer sizes and link capacities. Figure 3.5 shows results obtained using a link capacity of 250 cells/frame ($\approx$2.5Mbps) and ranges from 10 to 120 cell buffers (MQD$\approx$2$-$20ms). We can clearly see the good match from the SMP model.

3.2 LBC and VT plots

To fully investigate the behavior of the proposed model we present results obtained by the Leaky-Bucket Contour plot (LBC) [LNR94], as well as a variance-time plot for all our streams.

Figures 3.6 and 3.7 show additional results obtained analyzing our four streams of real and synthesized data. In the LBC plot of Figure 3.6, we assumed the buffer to be infinite (i.e. CLR=0) and the simulation records the maximum buffer occupancy for a range of link capacities; as in [CG96] we plot capacity versus the buffer-empty-

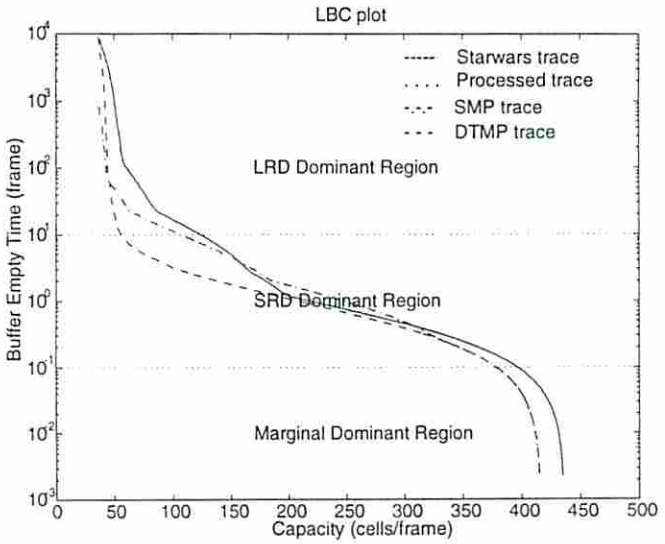

Figure 3.6: The leaky-bucket contour plot for the four traces.

time (T) which was found by $T = B/C$, in other words T is simply the MQD measured in frame time (i.e. MQD $= T/24$ in our case). Figure 3.6 shows that all streams are matching in part of the SRD dominant region, however, the DTMP - being a pure Markovian process - failed, as expected, to match the original stream in the LRD dominant region. The SMP with non-memoryless state holding times gave better results and appears to match some of the correlation structure of the original stream. It is important to remember, however, that the maximal delay and minimal unitization constraints found in real life networks restrict the allowable region for capacity allocation and buffer sizes. As shown in Figure 3.6, the lower and upper bounds separating the three regions is equivalent to a MQD$\approx$4.167ms and 416.67ms, respectively. We believe, therefore, that the proposed semi-Markov model mimics well the behavior of a real VBR traffic stream under realistic operational scenarios.

Figure 3.7 shows that the SMP is in fact more bursty in nature than its counterpart Markovian model, we can see that the stream generated by the SMP captured

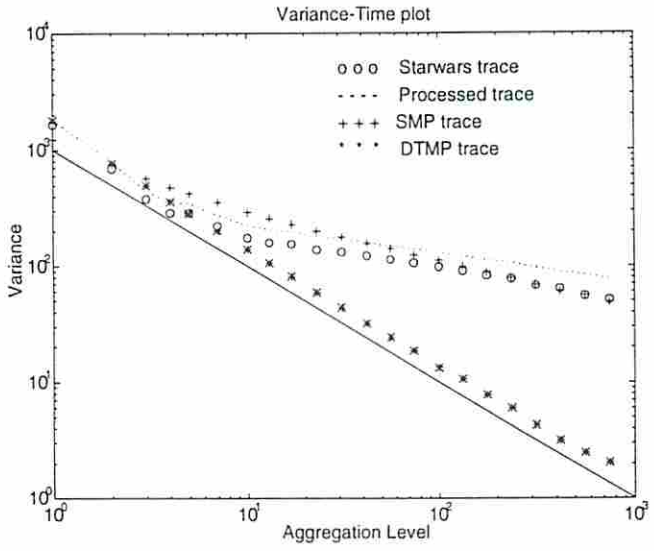

Figure 3.7: The variance-time plot for the four traces.

more of the correlation of the original stream. The DTMP, being inherently SRD, quickly looses its correlation as revealed by a slope of -1 (i.e. $H = .5$).

Finally, Figures 3.8 and 3.9 depict the actual traces. In Figure 3.8 we show a small segment of the original and processed trace showing the eight quantized levels. In Figure 3.9, we show the original, DTMP and SMP trace plotted at different time scales. It is clear that both the SMP and Star Wars trace appear to be more bursty and much more similar compared to the trace generated by the DTMP.

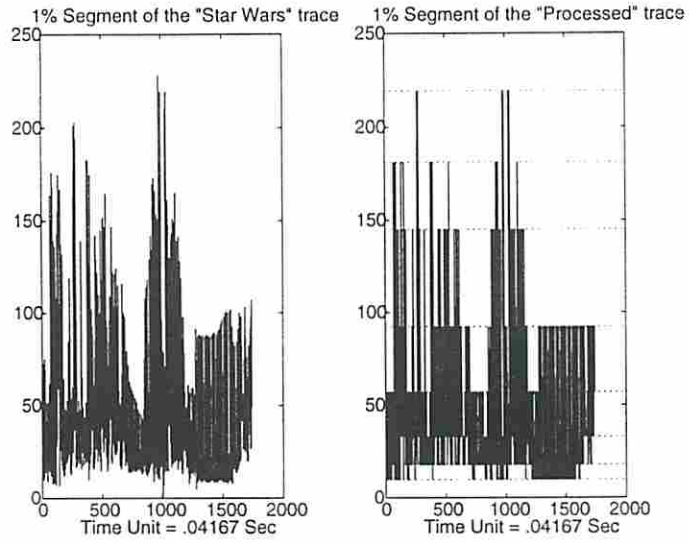

Figure 3.8: The eight quantized levels of the processed trace.

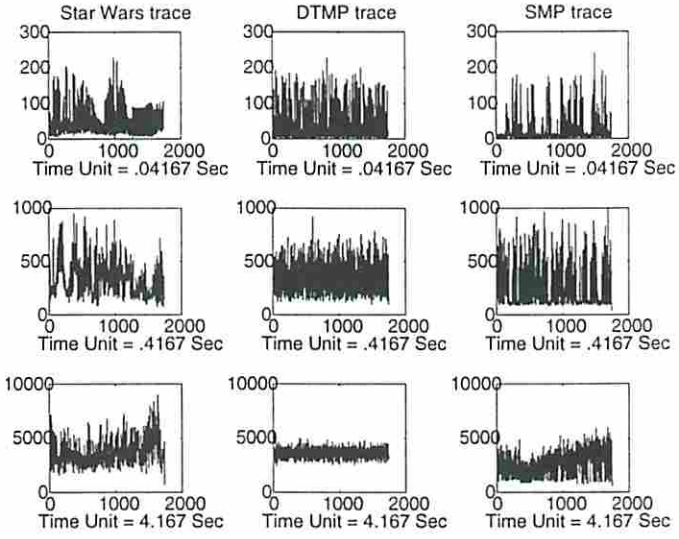

Figure 3.9: The three data traces shown at different levels of aggregation (Note: a Poisson process was used in the DTMP and SMP state).

Chapter 4

Conclusion

From results obtained by simulation, we conclude that the proposed clustering method enabled us to create accurate models for Variable-Bit-Rate (VBR) traffic modeling in computer networks. We used a similar algorithm to build semi-Markovian models and found it to be simple and efficient. Results showed that under realistic operational network scenarios, the impact of Long-Range Dependence (LRD) to determine the Cell-Loss-Rate (CLR) is not of practical importance. We showed that the use of well designed semi-Markovian models can give satisfactory results for computer network traffic modeling and performance evaluation.

We believe that there are still several issues that requires further investigation. Among those are how to formalize an efficient way to define the cluster sizes for best matching a given stream. We would also like to propose the use of different pdfs for state holding times, to see their goodness of fit by means of statistical tools such as the Quantile-Quantile plot [LK91]. We also propose using the data obtained by the clustering method to directly specify an empirical distribution for the modeling of state holding times. As described in Section 2.3, we propose the use of conditional probability for modeling state holding times.

We would also like to mention the possibility of utilizing the proposed clustering

method with the Hidden Markov Models (HMM) described in [RJ93]. We propose the use of the clustering method to identify the number of states in the Markov chain and corresponding average rate for each state, and then use the theory of HMM to find the transition probabilities.

Part II

Transport-layer Modeling

Chapter 5

Introduction

Simulation of computer networks is considered an efficient tool used in analysis and performance evaluation. In today's computer networks (e.g. the Internet), a range of users generate different types of traffic, and expect different types of response from the underlying network (i.e. each with a different set of rules to specify their QOS). In traffic modeling for simulation it is, hence, important to choose the best models that mimic the behavior of real users to ensure a correct analysis from a simulation point of view. This enables us to correctly predict the performance of a network prior to its deployment or redesign.

The concept of Long-Range-Dependence (LRD) initially reported in [LTWW94] opened new challenges in the engineering of computer network traffic. For many years researchers relied on Markovian models for network performance prediction, however, these models are inherently Short-Range Dependent (SRD). Other recent studies also revealed the self-similarity or "fractal" nature of streams collected over the Internet [Ryu98]. But what are the implications of LRD found in these streams from an engineering point of view, and why it is important to find new models that capture these features? We summarize the answer as follows:

1. The cell/packet loss probability in network queues decays faster when Markovian types of traffic models are used compared to comparable "fractal" models (i.e. with similar first and second order statistics).

2. The distribution of asymptotic queue size (i.e. maximum queue occupancy) decays faster in Markovian models.

These are some of the reasons encouraging researchers to gain a better understanding and find appropriate mathematical models that capture the behavior of real computer networks streams.

We just mentioned the effects of using inaccurate performance models (e.g. Markovian), the effects in a real network from a user perspectives are increased response times caused by either the very long queues, or, for reliable protocols (e.g. TCP), the retransmissions of dropped packets. For protocols such as ATM a degradation in the QOS (e.g. increased cell-loss rate) will result.

Having traffic models that correctly mimic real computer network traffic streams, allows us to generate a variety of synthesized streams that can be used in simulation; So that we may obtain correct analysis and performance evaluation of computer networks.

We classify the modeling and simulation of computer networks into two main categories: i) Link-level, and ii) Application or Transport-level. In the former, models are constructed and model-parameters are matched from a previously collected stream or group of streams. These models can be later used to generate synthesized streams in simulation. The latter approach is similar, however, the modeling is performed above the link-layer. In other words, while in the former method we usually analyze long traces of data at the packet or cell level, in the former approach we look at either user behavior, or more generally, higher-layer protocols (e.g. file size distributions, request lengths, etc.). In this part of the thesis, we focus on the latter

aspect, specifically, we analyze the behavior of the popular World Wide Web (also known as the Web or WWW).

5.1 Related Work

Several studies have attempted to understand how self-similarity arises in computer networks. In [PKC96] the ON/OFF model described in [WTSW95] was used simulate multiple client-server sessions emulating the behavior of Web traffic. [CB96] suggested that the reasons for traffic self-similarity can be attributed to the heavy-tailed nature of file size distributions available on the Web. [PF94] showed similar results modeling FTP bursts. Other results focused on the modeling via Cumulative-Distribution Functions (CDFs) estimated by empirical results, see [DJ91], [Edd96], [Mah97]. [Mah97] estimated several empirical CDFs to model HTTP traffic. The CDFs capture parameters of Web client/server behavior such as HTTP request/reply lengths, documents sizes and user think time. All of these studies require the use of a TCP algorithm to get packet-level results since the modeling is performed above the transport layer.

Before proceeding, we identify alternatives to model-driven simulations, and describe their major drawbacks:

1. Trace-driven simulation:

   One way to simulate real traffic is by using a trace or a group of traces collected from real computer networks. The three main drawbacks to this approach are:

   (a) A trace represents only a particular instance of "*history*".

   (b) The simulation is limited by the "*length*" of the trace.

   (c) We may need a large number of traces to accurately verify the simulation results.

2. Empirical-driven simulation:

   In this method, an empirical cumulative-distribution-function (CDF) is defined from a collection of real traffic traces, and hence used to generate random events in the simulation. This method too has its drawbacks which we identify briefly in the following two points:

   (a) In many cases the empirical CDF captures only first-order statistics (distribution). However, higher order statistics may be important.

   (b) Values of the random variables are "*bounded*" by the minimum and maximum values used to generate the empirical CDF. Large values, however, are not unlikely and have higher probability than previously thought (e.g. heavy-tailed distribution).

Even though empirical driven simulations can be considered of a somewhat more realistic approach than trace-driven simulations, the drawbacks mentioned suffice to motivate researchers to look for alternative methods in their simulations to insure realistic results.

The use of validated traffic models for simulation not only simplifies the simulation itself but can improve the simulation by giving more realistic results and permitting efficient and accurate network performance evaluation under a wide variability of scenarios. We briefly point out some advantages associated to this approach:

1. Virtually no limit in the *length* of the simulation.

2. A wider variability of scenarios can be simulated since traffic models can be adjusted to given input parameters with specific traffic characteristics.

Chapter 6

Issues and Traffic Model Proposed

In previous sections, we briefly mentioned that today's computer networks include a wide variety of heterogeneous traffic types. These traffic not only differ in nature, but users differ in the quality of service (QOS) they expect. It is of little practical use to propose a generic traffic model; we, therefore, focus on traffic generated by the popular World Wide Web (WWW or Web), with emphasis on the request arrival process.

6.1 Web Workload traffic (for request arrivals and volumes)

Traffic generated by the Web is considered the leading source of backbone network traffic found in today's Internet [Mah97]. WWW uses the Hypertext Transfer Protocol (HTTP) as its application layer protocol. HTTP uses the Transmission Control Protocol (TCP) as its transport layer protocol which is a *guaranteed* delivery protocol.

While users of this type of traffic tolerate delays (compared to other Internet

users such as users of real-time applications), there is little, if any, tolerance to loss of data. This implies that the network can be operated at wider ranges of utilization so long as data is not lost (this is, of course, guaranteed by TCP). We emphasize the possibility of operating the network at relatively high utilization levels (recall there is an inverse relationship between utilization and delay).

From analyzing a NASA and a Berkeley Web trace, [Ryu98] found that the request arrivals are well modeled by a superposition of fractal renewal processes (Sup-FRP). Compared to previous related work (e.g. [PKC96]), we show that ignoring second-order statistics in the Web arrival process will lead to inaccurate performance results in terms of response-time and packet loss.

6.2 The Superposition of Fractal Renewal Processes Model (Sup-FRP)

The Sup-FRP is a process generated from "M" independent and identically distributed (i.i.d.) fractal renewal processes (FRPs). Each FRP is defined by the following pdf:

$$p(t) = \begin{cases} \gamma A^{-1} e^{\frac{-\gamma t}{A}} & \text{for } t \leqslant A \\ \gamma e^{-\gamma} A^{\gamma} t^{-(\gamma+1)} & \text{for } t > A \end{cases} \tag{6.1}$$

With $1 < \gamma < 2$. The parameter A serves as a threshold between exponential behavior and power-law behavior of interarrival times. Figure 6.1 shows a graphical realization of the Sup-FRP.

In the Appendix we describe the algorithm used to generate the Sup-FRP arrival process in detail.

To fully describe the sup-FRP model we need the following three values to be

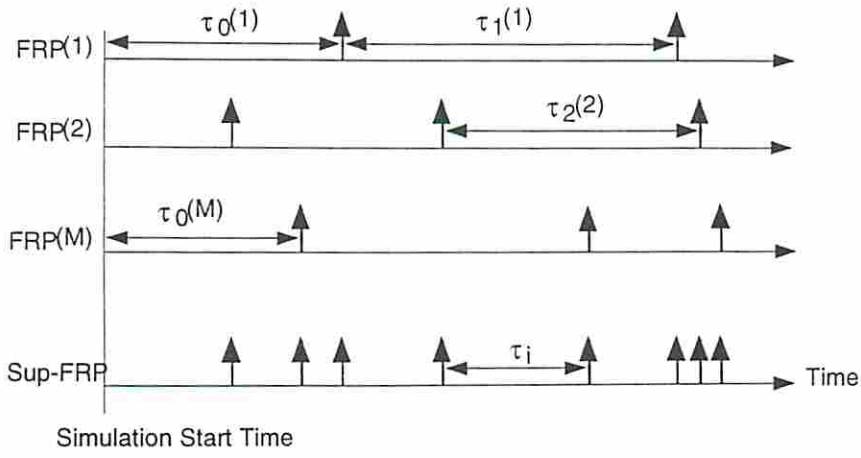

Figure 6.1: The Sup-FRP process shown graphically. Each FRP is i.i.d. (M=3 in the figure).

known:

1. γ :the shape of the pdf.

2. A :the cut-off value of the pdf.

3. M :the number of i.i.d. FRPs.

What is usually known, or can be estimated, from a given stream of traffic are the following three parameters:

1. H :the Hurst parameter defining the degree of self-similarity.

2. λ :the average arrival rate.

3. T_0^α :the onset-time (i.e. the level of aggregation where the fractal behavior starts, see [RL96], [RL98]).

There is a relationship between traffic and model parameters, the following set of equations defines this relationship. Details can be found in [RL96], [RL98]:

$$\gamma = 2 - \alpha \tag{6.2}$$

$$H = \frac{\alpha + 1}{2} \tag{6.3}$$

$$\lambda = M\gamma[1 + (\gamma - 1)^{-1}e^{-\gamma}]^{-1}A^{-1} \tag{6.4}$$

$$T_0^{\alpha} = 2^{-1}\gamma^{-2}(\gamma - 1)^{-1}(2 - \gamma)(3 - \gamma)e^{-\gamma}[1 + (\gamma - 1)e^{\gamma}]^2 A^{2-\gamma} \tag{6.5}$$

It is fairly simple to solve this system of equations to derive the Sup-FRP parameters. From Eq. (6.3) we can find α. Solving Eq. (6.2) we get γ. Next Eq. (6.5) yields A, and finally we obtain the value of M from Eq. (6.4).

6.3 The Sup-FRP Match the Arrival Process

From the analysis of two Web traces from NASA and Berkeley, Ryu [Ryu98] matches the IDC curve (Index of Dispersion for Counts, also known as the Fano factor F(T)) of the Web request arrival process and the Sup-FRP process. Figure 6.2 verifies this match. The IDC is defined as the variance of the number of arrivals in a given time window of width T divided by the mean number of arrivals in T. Note that for a Poisson point process (i.e. exponentially distributed interarrivals), the IDC curve value is 1 over the entire range of time scales. Recall that both the mean and variance of the Poisson process are identical, and the resulting aggregate process is still Poisson [Kle75].

We, therefore, use the Sup-FRP process to model the Web request arrival process.

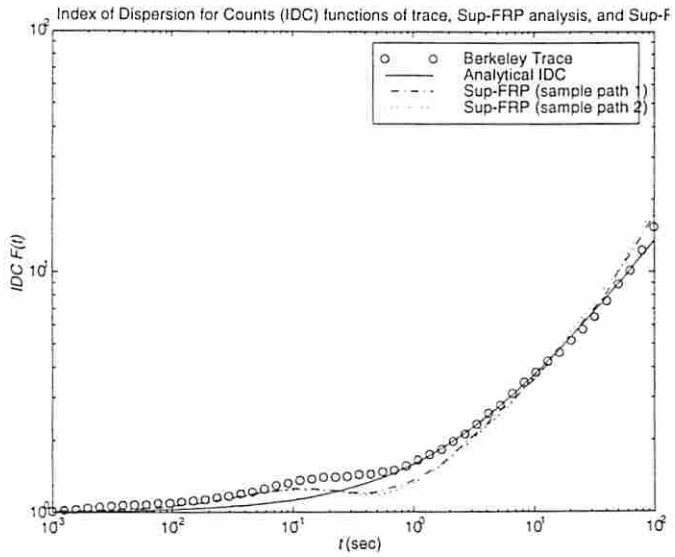

Figure 6.2: IDC match between the Sup-FRP and the Web request arrival process [Ryu98].

6.4 Heavy-Tailed Distributions: A Note

A random variable X follows a heavy-tailed distribution if the its complementary distribution (also known as the *survivor* distribution) has the form [AFT98]:

$$P[X > x] \sim x^{-\alpha}, \qquad \text{as } x \longrightarrow \infty,\ 0 < \alpha < 2 \tag{6.6}$$

An example is the Pareto distribution with probability density function:

$$f\left(x \mid \alpha, k\right) = \alpha k^{\alpha} x^{-(\alpha+1)} \qquad \alpha, k > 0,\ x \geqslant k \tag{6.7}$$

There are several properties associated with heavy-tailed distributions such as infinite mean and variance. The mean of the pareto pdf is:

$$E(X) = \int_{k}^{\infty} x f\left(x \mid \alpha, k\right) dx = \frac{\alpha k}{\alpha - 1} \tag{6.8}$$

The variance is:

$$Var(X) = E(X^2) - E(X)^2 = \frac{\alpha k^2}{(\alpha - 2)(\alpha - 1)^2} \tag{6.9}$$

Hence, for $\alpha \leqslant 2$, the distribution has infinite variance, and for $\alpha \leqslant 1$, the distribution has also infinite mean.

The value H (the Hurst parameter), and the parameter α of the Pareto pdf are related by the equation[PKC97],[AFT98]: $H = (3 - \alpha)/2$. Therefore, for LRD (i.e. $.5 < H < 1$) we require $1 < \alpha < 2$. For $\alpha > 2$ the process is SRD (i.e. $H < .5$).

Chapter 7

Network Model

The network model we use in our simulation is a fairly simple one. Figure 7.1 shows our simple two node topology.

The reason for this simple model is twofold; from a simulation point of view it allow us to investigate the behavior of the transport layer (TCP) and eliminate the effects of the routing protocol. From a modeling point of view it enables us to gain a better understanding of the effects of using the proposed arrival process. As shown in Figure 7.1, the request arrival process at node $G2$ initiates an FTP sessions. Note that an HTTP request may initiate several FTP sessions. The sup-FRP model proposed will be used to model the initiation (i.e. request arrival) of an FTP, not the arrival of an HTTP request. In other words, we model the FTP requests seen from the Web server at $G2$.

In our analysis we study the performance of the downstream traffic ($G2 \rightarrow G1$) for a wide range of buffer sizes. We also study network performance at different utilization levels (e.g. at different link speeds). By fixing the arrival process and changing the available capacity we investigate the effects of response time seen by the user.

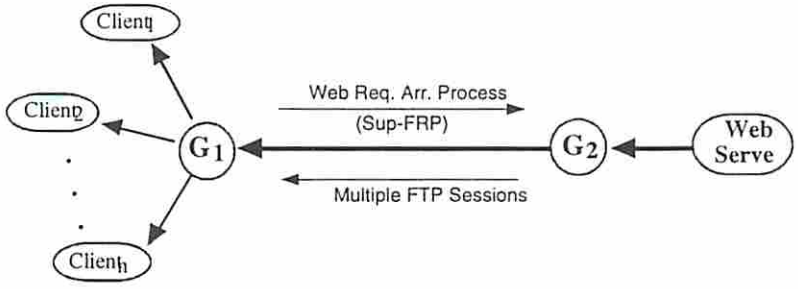

Figure 7.1: The two node simulation topology.

7.1 Simulation Environment

To study the performance impact of the proposed traffic model on TCP[1], we used the LBNL Network Simulator (ns) [MF98]. ns is an event-driven simulator derived from S. Keshav's REAL network simulator. We modified ns by adding an implementation of the Sup-FRP model.

To measure performance, we recorded throughput for each FTP session:

$$Throuput = \frac{File\ Size\ (bits)}{File\ Transmission\ Time\ (simulated\ seconds)} \tag{7.1}$$

As described in [PD96], the throughput can be thought of the *achieved* bandwidth, compared to the *available* bandwidth.

For each simulation run, we average throughput over all FTP sessions. For a given scenario (i.e. combination of arrival process and file size distribution), we find the averaged throughput at different levels of network utilization ρ in the range $[0.1, 0.9]$. Utilization is found by:

$$\rho = \frac{\lambda F}{C} \tag{7.2}$$

[1]Our focus here is to verify the general impact of the arrival process on network performance. We will, therefore, use only one of the available TCP flavours, namely, "Tahoe" TCP.

where, λ = Average arrival rate (requests/sec), F = Average file size (bits), and C = Link capacity (bps).

To vary ρ, we adjusted the value of C and fixed λ, and F for a given run.

Chapter 8

Simulation Results

We performed several experiments for different arrival processes (e.g. Exponential, and Sup-FRP) and file size distributions (e.g. Exponential, and Pareto). For a given experiment, we kept the average arrival rate and mean file size equal to ensure a parsimonious comparison, in other words, whether the interarrival request was exponentially distributed or followed the Sup-FRP process, we used the same average arrival rate (for the Sup-FRP process we adjusted the onset-time $(T_0^{\alpha}) \approx [.1, .3]$, and the Hurst parameter $(H) \approx .8$). The same for file sizes. Table 8.1 summarize values of link capacity (C) for different values of utilization (ρ). We used an average arrival rate (λ) of 10 arrivals/sec[1], and average file size (F) of 9.375KB. The $G2 \rightarrow G1$ link-delay was fixed at 20ms. In each simulation run, we generated 3000 arrivals (i.e. approximately 5 simulated minutes).

Utilization "ρ"	.1	.2	.3	.4	.5	.6	.7	.8	.9
Capacity "C" (Mbps)	7.5	3.75	2.5	1.875	1.5	1.2	1.07	.9375	.8333

Table 8.1: Network Utilization and equivalent Link Capacity.

[1]The low arrival rate (e.g. $\lambda \cong 10$) was motivated to lower the number of superimposed FTP sessions. We believe that similar results consistent with our findings will be achieved for larger-scale simulations (e.g. $\lambda \cong 1000$).

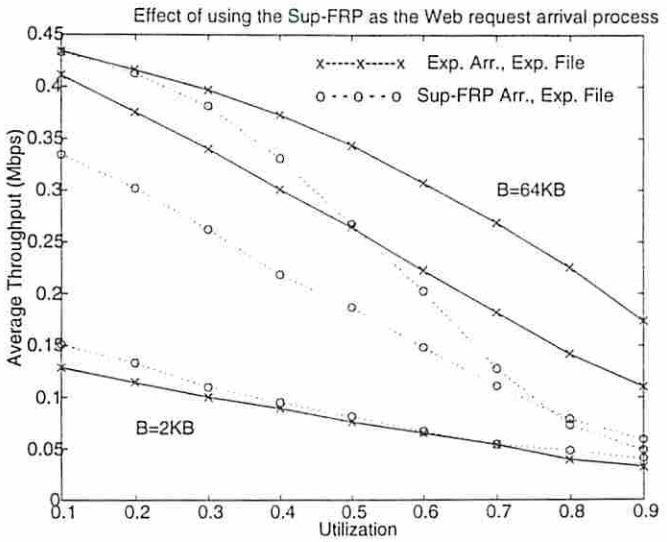

Figure 8.1: Effects from correlated arrivals get pronounced as the buffer size increrase (we used B=2KB, 4KB, and 64KB).

In Figure 8.1 we see the effect of using correlated arrivals. We used two different arrival processes, one uncorrelated (exponentially distributed interarrivals) and the other correlated (Sup-FRP arrivals). To evaluate the effects of the arrival process on throughput and eliminate effects due to heavy-tailed file size pdf, we used exponentially distributed file sizes. We performed several simulation runs with buffer size of 2KB, 4KB, and 64KB. As the buffer size increase, we observe a pronounced effect of Sup-FRP on average throughput. It is interesting to see that strongly correlated arrivals not only lowered the achieved throughput, but had a stronger effect at higher utilization levels; we attribute this result to: a) increased buffer occupancy at higher utilization, and b) increased number of packets dropped, hence, increased transmissions times due to the retransmission of lost packets.

In the next experiment we investigated the performance impact using both a correlated arrival process, and a heavy-tailed file size distribution. As reported in [CB96], traffic self-similarity may be attributed to the heavy-tailed nature of file

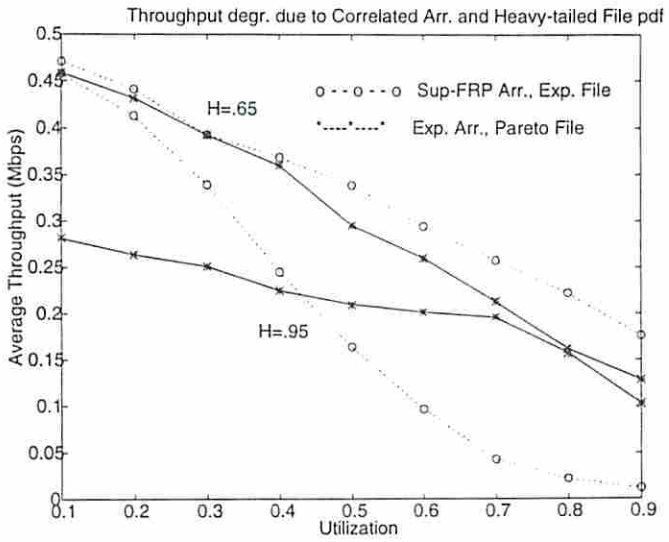

Figure 8.2: Effects of changing the Hurst parameter of the Arrival process and File size pdf on Throughput.

sizes found in the Web. Even though several studies (e.g. [PKC96]) relied on the findings of [CB96], we emphasize the importance of characterizing the nature of files *requested* (in contrast to files *found* or observed for multiple requests). In [AFT98] it is argued that the presence of caching in the Web has the effect of making the set of transmitted files relatively insensitive to the set of files requested, and distributionally similar to the set of available files.

In Figure 8.2, we compare the performance impact due to a heavy-tailed file size distribution and correlated request arrivals for different values of H (the Hurst parameter).

We observe the performance impact when file sizes follow a heavy-tailed distribution (e.g. Pareto), and notice that the effect is mainly at lower utilization levels (ρ in the range $[0.1, 0.5]$). The arrival process, as previously observed has a predominant effect at higher utilization levels (ρ in the range $[0.5, 0.9]$). Figure 8.3 summarizes the main results (for $H = .8$).

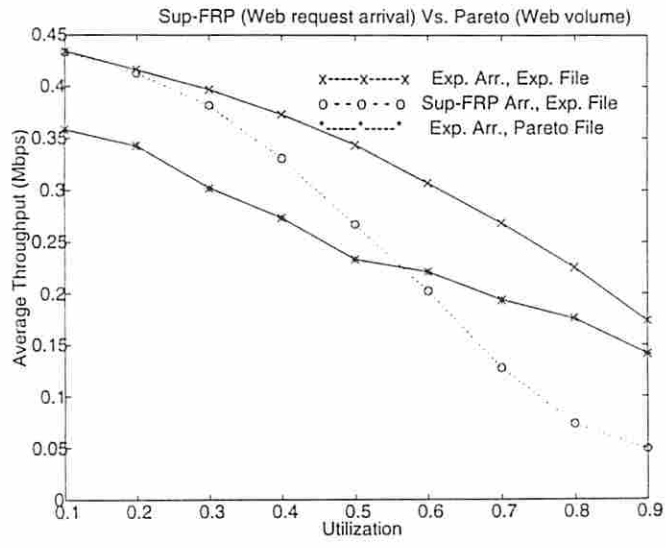

Figure 8.3: File size distribution impact performance at low utilization levels. The arrival process impact performance at high utilization level.

Since the observed Web request arrival process is well matched by the Sup-FRP process [Ryu98], and previous studies reported that Web file sizes are well modeled by the pareto pdf [PKC96], [Mah97], [AFT98]; we analyze the performance impact using the Sup-FRP process to simulate the Web request arrival process, and the Pareto pdf to simulate Web volumes. Figure 8.4 reveals an enormous degradation in performance due to both phenomena.

In Figure 8.5, we plot the average FTP session delay recorded. The Sup-FRP resulted in longer delays compared to delays predicted by both an uncorrelated arrival process (e.g. exponentially distributed interarrivals), and heavy-tailed file size distribution (e.g. Pareto). Our intuition attributes this observation to: a) overall increased queue lengths with the Sup-FRP arrival process, and b) increase in the number of dropped packets for most of the FTP sessions with the Sup-FRP arrival process. We also argue that the lower delays observed at higher utilization for Pareto distributed file sizes may be due to the high variability in the file sizes

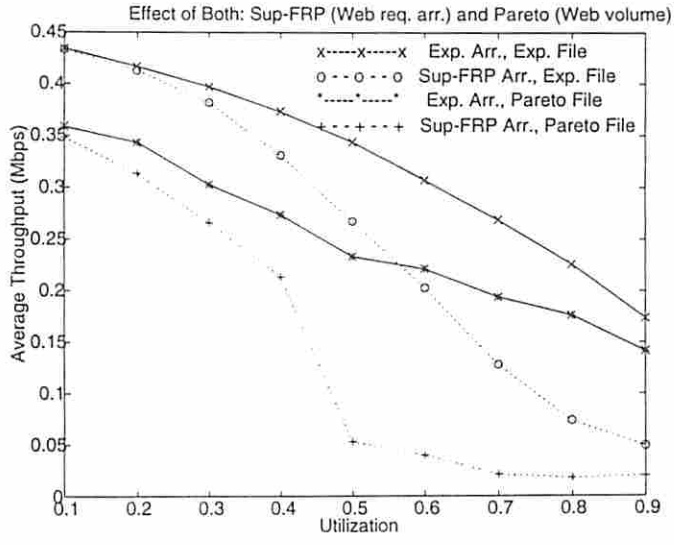

Figure 8.4: Both a strongly correlated arrival process and heavy-tailed file pdf result in high performance degradation.

and increased number of small files.

To investigate the results of Figure 8.5 we plot the delay variance for all FTP sessions. Figure 8.6 shows the variance plotted on a log scale. We observe that using a heavy-tailed file size pdf introduces high variability at lower utilization levels, compared to the Sup-FRP process which affects utilization at higher levels.

In Table 8.2, we summarize our results. We describe the effects of correlated arrivals (e.g. Sup-FRP) and the heavy-tailed distributed file sizes (e.g. Pareto) on throughout and response time. We compare the effects of both phenomena with uncorrelated arrivals and exponentially distributed file sizes.

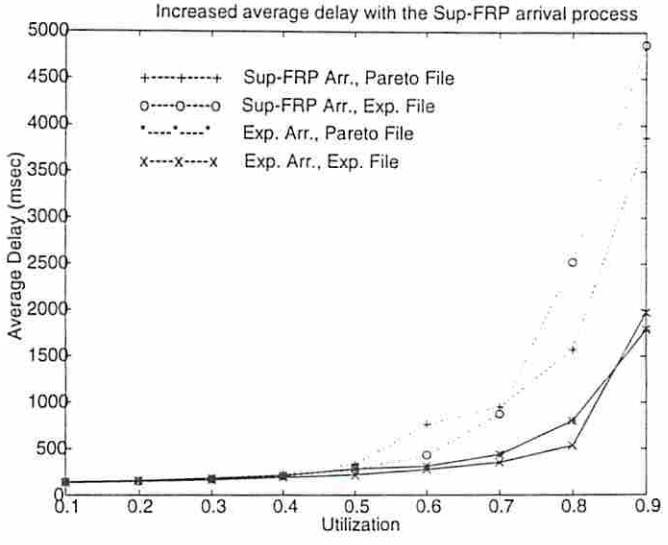

Figure 8.5: Average file transmission time increase with the Sup-FRP arrival process.

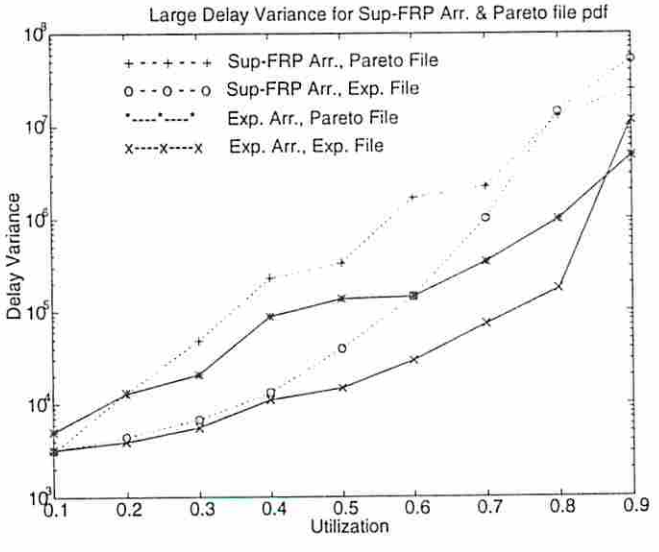

Figure 8.6: Variance in file transmission time increase with both the Sup-FRP arrival process and Heavy-tailed file pdf.

<<Effects>>	Correlated Arrivals (Sup-FRP)	Heavy-Tailed File Dist. (Pareto)
Average Throughput	degradation at higher util. levels ($\rho \cong [0.5, 0.9]$).	degradation at lower util. levels ($\rho \cong [0.1, 0.5]$).
Average Response Time	increases faster from lower utilization levels ($\rho \cong .65$).	only little incr. at lower util. due to incr. no. of small files.
Response Time Variance	increased at higher utilization levels ($\rho \cong [0.5, 0.9]$).	higher over the entire range of utilization.

Table 8.2: Effects of correlated arrivals and heavy-tailed file size pdf on performance: Summary of Results.

Chapter 9

Conclusion

We analyzed the performance impact in Internet Web traffic due to correlated arrivals and heavy-tailed file size distribution. We used the Sup-FRP process proposed in [Ryu98] for modeling the Web request arrival process and compared the performance impact with uncorrelated arrivals (i.e. exponentially distributed inter-arrivals) and heavy-tailed file size pdf (e.g. Pareto). From several experiments, we show that ignoring the correlation found in the Web request arrival process will lead to inaccurate performance analysis.

As reported in [PKC96], [AFT98], and [Mah97], we verified that the heavy-tailed nature of files found in the Web lead to a performance degradation, furthermore, we show that the network performance impact due to heavy-tailed file size distribution is mostly at low utilization level. We also show that correlated Web requests (e.g. Sup-FRP) affects performance at higher utilization level.

Since Web users are tolerant to delays (compared to users of real-time application), we believe it is not unlikely to operate networks at high utilization levels; Therefore, we believe that the arrival process has an impact on performance and needs to be well characterized.

Another issue we believe deserves investigation is to compare transmission time

distributions with the Sup-FRP arrival process, and both exponential and Pareto file size pdfs. [AFT98] shows that there does not seem to be strong sample correlation between file sizes and transmission times. We, therefore, believe that the arrival process may be the reason for the observed heavy-tailed nature in the distribution of transmission times. In addition, in order to fully validate the Sup-FRP model, we propose to compare our results to empirical studies previously reported (e.g. [Mah97], [DJ91]).

Chapter 10

Appendix: Derivation of the Sup-FRP

We present a complete derivation for the Sup-FRP process described in Section 6.2.

The interarrival time of each Fractal-Renewal-Process (FRP) of Figure 6.1 is defined by the pdf:

$$p(t) = \begin{cases} \gamma A^{-1} e^{\frac{-\gamma t}{A}} & \text{for } t \leqslant A \\ \gamma e^{-\gamma} A^{\gamma} t^{-(\gamma+1)} & \text{for } t > A \end{cases} \tag{10.1}$$

We find the cumulative-distribution-function (CDF) for the interarrival time:

for $t \leqslant A$

$$F(t) = \int_0^t \gamma A^{-1} e^{\frac{-\gamma t}{A}} dt = 1 - e^{\frac{-\gamma t}{A}} \tag{10.2}$$

similarly for $t > A$

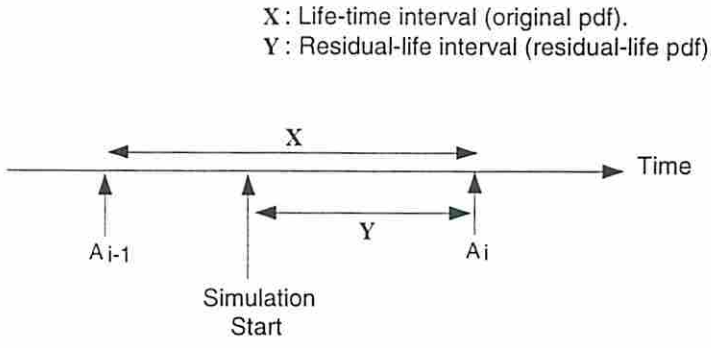

Figure 10.1: Life-time and Residual-life interval.

$$F(t) = \int_0^A \gamma A^{-1} e^{\frac{-\gamma t}{A}} dt + \int_A^t \gamma e^{-\gamma} A^{\gamma} t^{-(\gamma+1)} dt = 1 - e^{-\gamma} A^{\gamma} t^{-\gamma} \tag{10.3}$$

Using the inverse method and solving both equations we get:

$$T = \begin{cases} \frac{-A}{\gamma} \ln U & U \geq e^{-\gamma} \\ A e^{-1} U^{\frac{-1}{\gamma}} & U < e^{-\gamma} \end{cases} \tag{10.4}$$

Since the continuous pdf of (10.1) is not the exponential pdf (i.e. is not memory-less), we need to derive the distribution of the *residual-life* to model the first interval τ_0 of each FRP, see Figures 6.1 and 10.1. From [Kle75], the pdf of the residual-life is defined as follow:

$$p_{residual}(t) = \frac{1 - F(t)}{E(t)} \tag{10.5}$$

$$E(t) = \int_0^A t\gamma A^{-1} e^{\frac{-\gamma t}{A}} dt + \int_A^{\infty} t\gamma e^{-\gamma} A^{\gamma} t^{-(\gamma+1)} dt = \frac{A}{\gamma(\gamma - 1)} (e^{-\gamma} + \gamma - 1) \tag{10.6}$$

the pdf of the residual life is then:

$$p_{residual}(t) = \begin{cases} \frac{\gamma(\gamma-1)e^{\frac{-\gamma t}{A}}}{A(e^{-\gamma}+\gamma-1)} & \text{for } t \leqslant A \\ \frac{\gamma(\gamma-1)e^{-\gamma}A^{\gamma}t^{-\gamma}}{A(e^{-\gamma}+\gamma-1)} & \text{for } t > A \end{cases} \tag{10.7}$$

As before, we find the CDF for the residual-life interval:

for $t \leqslant A$

$$F_{residual}(t) = \int_0^t \frac{\gamma(\gamma-1)e^{\frac{-\gamma t}{A}}}{A(e^{-\gamma}+\gamma-1)} = \frac{\gamma-1}{e^{-\gamma}+\gamma-1}(1-e^{\frac{-\gamma t}{A}}) \tag{10.8}$$

similarly for $t > A$

$$\begin{aligned} F_{residual}(t) &= \int_0^A \frac{\gamma(\gamma-1)e^{\frac{-\gamma t}{A}}}{A(e^{-\gamma}+\gamma-1)}dt + \int_A^t \frac{\gamma(\gamma-1)e^{-\gamma}A^{\gamma}t^{-\gamma}}{A(e^{-\gamma}+\gamma-1)}dt \\ &= \frac{1}{e^{-\gamma}+\gamma-1}\left(\gamma-1+e^{-\gamma}-\gamma e^{-\gamma}A^{\gamma-1}t^{1-\gamma}\right) \end{aligned} \tag{10.9}$$

Using the inverse method and solving both equations we get:

$$T_{residual} = \begin{cases} -\gamma^{-1}A\ln[U+(U-1)(\gamma-1)^{-1}e^{-\gamma}] & V \geq 1 \\ AV^{\frac{1}{1-\gamma}} & V < 1 \end{cases} \tag{10.10}$$

where,

$$V = \frac{1+(\gamma-1)e^{\gamma}}{\gamma}U \tag{10.11}$$

Hence, as shown in Figure 6.1, to generate Sup-FRP arrivals, we use $T_{residual}$ Eq. (10.10) to find $\tau_0^{(i)}$, and T Eq. (10.4) to find $\tau_j^{(i)}$, $\forall i = 1, 2, ..., M$ and $j > 0$.

Bibliography

[AFT98] R. J. Adler, R. E. Feldman, M. S. Taqqu, "A Practical Guide to Heavy Tails," Birkhauser, pp. 3-25, 1998.

[BSTW95] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-Range Dependence in Variable-Bit-Rate Video Traffic," IEEE Trans. Commun., vol. 43, no. 2/3/4, pp. 1566-1579, Feb./March/April 1995.

[CB96] M. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: Evidence and possible causes," In Proceedings of the ACM SIGMETRICS Conference on Measurement & Modeling of Computer Systems, pp. 160-169, Philadelphia, PA, May 1996.

[CG96] C.-H. Chou and E. Geraniotis, "Performance Prediction and Resource Allocation for Long-Range Dependent Traffic in ATM Networks," Proc. of the 30th Annual Conference on Information Sciences and Systems, pp.198-204, March 1996.

[DH73] R. O. Duda and P. E. Hart, "Pattern Classification and Scene Analysis," John Wiley & Sons, 1973.

[DJ91] Peter B. Danzig and Sugih Jamin, "tcplib: A library of TCP internetwork traffic characteristics," Technical Report USC-CS-91-495, Com-

puter Science Department, University of Southern California, Los Angeles, CA, 1991.

[Edd96] Rusty Eddy, "HTTP analysis of IP level traces, Feb. 1996, Available at http://catarica.usc.edu/eddy/http-traffic/http-traces.html.

[GW94] M. Garrett and W. Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic," Proc. ACM SIGCOM '94, pp. 269-280, Sep. 1994.

[Kle75] Leonard Kleinrock, "Queueing Systems, Volume I: Theory," John Wiley & Sons, 1975.

[LK91] A. M. Law and W. D. Kelton, "Simulation Modeling and Analysis," Second Edition, McGraw-Hill, pp. 375-379 ,1991.

[LNR94] D. M. Lucatoni, M. F. Neuts, and A. R. Reibman, "Methods for Performance Evaluation of VBR Video Traffic Models," IEEE/ACM Trans Networking, vol. 2, no. 2, pp. 176-180, April 1994.

[LTWW94] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," IEEE/ACM Trans. Networking, vol. 2, no. 1, pp.1-15, Feb. 1994.

[LTWW95] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "Self-Similarity in High-Speed Packet Traffic: Analysis and Modeling of Ethernet Traffic Measurements," Statistical Science, vol. 10, no. 1. pp. 67-85, 1995.

[Mah97] Bruce A. Mah, "An Empirical Model of HTTP Network Traffic," Proceedings of INFOCOM '97, Kobe, Japan, April 1997.

[MF98] S. McCanne and S. Floyd, "ns - Network Simulator," Available at http://www-mash.cs.berkeley.edu/ns/.

[Nel95] R. Nelson, "Probability, Stochastic Processes, and Queueing Theory," Springer-Verlag, pp. 352-356, 1995.

[PD96] Larry L. Peterson & Bruce S. Davie, "Computer Networks, A System Approach," Morgan Kaufmann Publishers, 1996.

[PF94] V. Paxson and S. Floyd, "Wide Area Traffic: The Failure of Poisson Modeling," In Proc. ACM SIGCOMM '94, pp. 257-268, 1994.

[PKC96] K. Park, G. T. Kim, and M. E. Crovella, "On the relationship between file sizes, transport protocols, and self-similar network traffic," in Proceedings of the Fourth International Conference on Network Protocols (ICNP '96), pp. 171-180, October, 1996.

[PKC97] K. Park, G. T. Kim, and M. E. Crovella, "On the Effect of Traffic Self-Similarity on Network Performance," CSD-TR 97-024, Dept. of Computer Sciences, Purdue University, July 1997.

[PVTF92] W. H. Press, W. T. Vetterling, S. A. Teukolsky, B. P. Flannery, "Numerical Recipes in C," Second Edition, Cambridge University Press, 1992.

[RE96] B. K. Ryu and A. Elwalid, "The importance of Long-Range Dependence of VBR video traffic in ATM traffic engineering: Myths and Realities," In Proc. ACM SIGCOMM, San Francisco, CA, 1996.

[RJ93] L. Rabiner and B. H. Juang, "Fundamentals of Speech Recognition," Prentice Hall, 1993.

[RL96] B. K. Ryu and S. B. Lowen, "Point process approaches to the modeling and analysis of self-similar traffic: Part I - Model constriction. In Proc. IEEE INFOCOM '96, San Francisco, CA, 1996.

[RL98] B. K. Ryu and S. B. Lowen, "Point process models for self-similar network traffic, with applications" Stochastic Models (M. Neuts, Ed.), vol. 14, Nr. 3, pp. 735-761, 1998.

[Ryu98] Bo Ryu, "Modeling and Simulation of Broadband Satellite Networks: Part II - Traffic Modeling," submitted to IEEE Communications Magazine, 1998.

[SS93] P. Skelly and M. Schwartz, "A Histogram-Based Model for Video Traffic Behavior in an ATM Multiplexer," IEEE/ACM Trans. Networking, vol. 1, no. 4, pp. 446-459, Aug. 1993.

[WTSW95] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson, "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level," In Proc. ACM SIGCOMM '95, pp. 100-113, 1995.