ABSTRACT

Title of Thesis: TRAFFIC MODELS FOR HYBRID SATELLITE-TERRESTRIAL NETWORKS

Degree Candidate: Bradley Alfred Barrett

Degree and year: Master of Science, 1999

Thesis directed by: Professor John S. Baras
Institute for Systems Research

While Hybrid Satellite-Terrestrial Networks (HSTNs) have become a popular method of providing internet connectivity, network dimensioning and performance prediction problems in these networks—as in their terrestrial counterparts—have remain largely unsolved. A key hindrance to the resolution of these issues has been accurate, tractable traffic models. While a number of rather complex models have been proposed for terrestrial network traffic, these have not been evaluated against HSTN traffic. And further, recent studies have questioned whether these more complex models, while statistically better fits, really provide significantly better performance prediction.

We examine the question of how to model HSTN traffic for network dimensioning and performance prediction, and in particular, how far ahead into the future a traffic model can be expected to accurately function. We investigate these issues by directly comparing four of the most likely candidate statistical distributions—the

exponential, log-normal, Weibull and Pareto. These distributions are fit to two key traffic parameters from real HSTN traffic traces (connection interarrival times and downloaded bytes), and their relative fits are compared using statistical techniques. We further compare traffic models built using these distributions in a simulated environment; comparing performance predictions (over a number of metrics) obtained from these models to the actual results from our real-world traffic traces.

TRAFFIC MODELS FOR HYBRID SATELLITE-TERRESTRIAL NETWORKS

By

Bradley Alfred Barrett

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering
1999

Advisory Committee:
Professor John S. Baras, Chair
Associate Professor Mark A. Austin
Associate Research Scientist M. Scott Corson

Dedication

To my father, for sharing his love of engineering; and my mother—my first teacher.

Acknowledgements

I am indebted to a number of people and organizations for their assistance with this work. This study could never have been undertaken without the generous access granted by Hughes Networks Systems to collect data from their commercial network. Their support of this work went far beyond the usual financial backing, and, to me, helped make this project such a successful example of the benefits of industrial and academic collaboration. Rod Ragland, at Hughes Networks Systems, provided invaluable assistance in collecting the traces used in this work, and in answering all my questions about the NOC architecture. He even gave up a night when he could have been home with his family, to let me into the building and help me take the traces. My advisor, Dr. John Baras was also instrumental in this work from start to finish, helping me formulate the problem, refine my approach, and prepare it for presentation. And as the director of the Center for Satellite and Hybrid Communication Networks (CSHCN), he is also principally responsible both for arranging this unique research opportunity and providing the excellent research environment in which I was able to work.

I am also very grateful for the assistance of Manish Karir, manager of the Systems Engineering and Integration Laboratory, who helped me assemble the hardware used to collect the traces, and set up the large amount of disk space needed to work with them. He and a number of other fellow graduate students: Steve Payne, Majid Raissi-Dehkordi, Mingyan Liu and Vijay Bharadwaj at various times provided feedback, insight, a brain to pick, a wall off of which to bounce ideas, or a joke to break up a long day’s work. My thanks to them all.

The research reported in this thesis was partially supported through contracts from Hughes Network Systems, Lockheed Martin Global Telecommunications and the Center for Satellite & Hybrid Communication Networks, a NASA Commercial Space Center (CSC), under NASA Cooperative Agreement NCC3-528. Their support is gratefully acknowledged.

Table of Contents

Chapter 1 – Introduction..1

1.1 Motivation and Significance..1

1.2 Contributions of this Thesis...3

1.2.1 Outline 4

Chapter 2 – Hybrid Internet...6

2.1 Background 6

2.2 Technical Issues...8

2.3 Hybrid Internet Traffic Modeling...8

Chapter 3 – Network Traffic Models..10

3.1 Traffic History: The Poisson Model..10

3.2 Toward Modern Traffic Models 12

3.3 Long Range Dependence...13

3.4 Log-normal Distribution..15

3.5 Weibull Distribution...16

3.6 Pareto Distribution...17

3.7 Fit Comparison 18

Chapter 4 – Traffic Traces and Modeling..20

4.1 DirecPC™ Trace(s) 20

4.1.1 Trace Summaries 21

4.1.2 Trace Validation..23

4.2 Connection Interarrivals 25

4.3 Bytes downloaded (per connection) 31

4.4 Summary..37

Chapter 5 – HSTN Testbed Simulations 39

5.1 Opnet HSTN Testbed 39

5.1.1 Testbed Hybrid Gateway..40

5.1.2 Testbed Satellite Gateway..41

5.1.3 Testbed Client/Server 42

5.2 Modeling Studies on the HSTN Testbed 42

5.2.1 Test Method 43

5.2.2 Performance Metrics...43

Chapter 6 - Conclusions..58

6.1 Conclusions...58

6.2 Future Work..59

Bibliography 60

Table of Figures

Figure 2.1 – Typical HSTN Configuration...7

Figure 4.1 – Connection Interarrival Autocorrelations (for traces 1,2,3)..........................24

Figure 4.2 - Response Size Autocorrelations (for traces 1,2,3).......................................25

Figure 4.3 – Distributions of interarrivals (linear scale) (traces 1,2,3)............................26

Figure 4.4 – Distributions of interarrivals (semilog scale) (traces 1,2,3) 27

Figure 4.5 - Interarrival distribution of Trace 1...28

Figure 4.6 - Interarrival distribution of Trace 2...29

Figure 4.7 - Interarrival distribution of Trace 3...30

Figure 4.8 - Download size distributions..32

Figure 4.9 - Trace 1 download sizes (bytes)...33

Figure 4.10 - Download size fits for trace 1 35

Figure 4.12 - Download size fits for trace 2 36

Figure 4.13 - Download size fits for trace 3 37

Figure 5.1 – Opnet HSTN Testbed Network 40

Figure 5.2 - Peak Throughput, Predicted vs. Actual (bytes/s)..45

Figure 5.3 - Average Throughput, Predicted vs. Actual (bytes/s) 47

Figure 5.4 - Maximum SGW Queue Length, Predicted vs. Actual (bytes).......................49

Figure 5.5 - Average SGW Queue Length, Predicted vs. Actual (bytes)..........................51

Figure 5.6 - Peak Number of Connections on HGW, Predicted vs. Actual 53

Figure 5.7 - Average UDP Packet Delivery Delay, Predicted vs. Actual (sec)................55

Chapter 1 – Introduction

1.1 Motivation and Significance

With the explosive growth of the Internet, bandwidth demand has for the past several years exceeded supply at the network edge, particularly for home and small business users. Conventional remedies to the 56 – 128kbps ceiling of analog modems and Integrated Services Digital Networks (ISDN)—long promised technologies such as cable modems and Digital Subscriber Lines (DSL)—have languished in their deployment for large sectors of users. Deployment of these technologies has, in addition, been hampered by equipment and infrastructure costs as well as uncertainties in the market and business case evaluations. As a result the "power" home Internet user, the "telecommuter," and the small business have faced a paucity of connectivity options available to them in the significant price and performance gap between ISDN (128kbps) and T-1 (1.5Mbps) services. And while these broadband technologies have begun to be more widely deployed in the last year, a number of customers, in the U.S. and especially abroad, will remain outside their service areas for the near future.

While a number of wireless alternatives have been demonstrated, only one has been widely deployed to date: hybrid satellite-terrestrial networks (HSTNs). The

DirecPC™ system, developed by Hughes Network Systems, is the principal example of this technology. The system provides a downstream bandwidth to its users of 400kbps or more via a satellite channel. Upstream bandwidth is provided over terrestrial telephone lines via a conventional voiceband modem. However, because classic TCP/IP is not well suited to satellite channels, it, like all HSTNs, must overcome several technical hurdles to provide a comparable quality of service. This adds to the complexity of the gateways that are responsible for forwarding traffic over the satellite. And while the problems—such as long satellite link delay and connection fairness—have been studied and addressed with solutions like connection splitting (also called "spoofing")[1] and flow control; performance prediction and network dimensioning problems have remained unsolved, in part because traffic models for HSTN networks have not been fully studied.

Wide area network (WAN) traffic models have also changed rapidly in recent years, principally due to Internet traffic. The "burstiness" of packet arrivals, at all time scales, has found expression in fractal mathematical models for traffic. To incorporate the required long range dependence, a number of different statistical distributions have been suggested for both interarrival times and durations. Recent studies have indicated, however, that although these more complex models more accurately characterize source and network traffic, when that source traffic is fed through a queueing system the resulting performance predictions (based on these more complex models) may offer only limited additional insight or accuracy over those obtained from simpler, more traditional models. At the present state of analysis, measurement and experimentation, we do not

[1] We prefer the term "connection-splitting" as it better describes what takes place, and does not also refer to a method of security breach used by hackers, as does "spoofing".

have a clear understanding of the implications or benefits of these more complex models for network control, performance evaluation and resource allocation.

The question then arises as to how to model HSTN traffic for performance prediction, and in particular, how far ahead into the future a traffic model can be expected to accurately function. We investigate these issues by directly comparing four of the most likely candidate statistical distributions—the exponential, log-normal, Weibull and Pareto. These distributions are fit to two key traffic parameters from real HSTN traffic traces (connection interarrival times and downloaded bytes), and their relative fits are compared using statistical techniques. We further compare traffic models built using these distributions in a simulated environment; comparing performance predictions (over a number of metrics) obtained from these models to the actual results from our real-world traffic traces.

While terrestrial Internet traffic has been studied extensively recently [Pax94] [WTSW97], it has typically been studied over rather small networks. There are now several World Wide Web sites with traffic traces from terrestrial Internet available to researchers [BU] [ITA]. However these traces are taken from networks that are neither satellite/wireless, nor commercial size. Our research and investigation is among the very few (but widely required) studies that used actual data from a large commercial service providing hybrid Internet over satellites.

1.2 Contributions of this Thesis

The basis of this work is a series of traffic traces taken from a large, commercial HSTN known as the DirecPC system. These traces, collected by the author, are used as the heart of a comparative study of HSTN traffic models undertaken on a unique HSTN simulation testbed constructed by the author in the OPNET discrete event network simulator. This thesis presents the most complete study to date of HSTN Internet traffic and its most suitable models, and in addition offers substantial insight into the utility (and limitations) of performance prediction via statistical traffic models in the HSTN environment.

1.2.1 Outline

We examine the applicability of various wide-area traffic model distributions in the setting of hybrid satellite-terrestrial networks-- with a particular eye toward two issues: first, what we term the "marginal utility" (that is, the additional benefit resulting from their use) of newer "self-similar" models over traditional ones; and second, the "lifetime" of a model (by this we mean how long it remains valid after being fit).

We begin with a review of the basics of hybrid satellite-terrestrial networks with an emphasis on their unique technical characteristics in Chapter 2. The general HSTN topology is presented, along with an explanation of the function of the key elements of the HSTN Network Operations Center (NOC), including connection-splitting and flow control.

Chapter 3 reviews the traffic models utilized in this study, and additionally covers results of recent comparative traffic model studies. Beginning from a historical perspective we trace the development of modern data network traffic models and statistical distributions. We present also the parameter estimation techniques for each distribution used in this study.

Three large traces were collected from an HSTN NOC, and used as the reference for traffic model comparison. Chapter 4 details the traffic traces used for the study, and evaluates the statistical fit of our models to these traces. Comparisons between the models and their fits are provided.

An HSTN network testbed was created in the OPNET discrete event simulator. Chapter 5 describes this simulation testbed and presents the results of our traffic model evaluations and comparisons performed with it.

Conclusions are presented in the final chapter.

Chapter 2 – Hybrid Internet

We shall describe Hughes Networks Systems (HNS) DirecPC as our example of a hybrid satellite-terrestrial Internet service. However, many of the technical issues we discuss are applicable to all systems providing TCP/IP network connectivity via geostationary satellite(s).

2.1 Background

The DirecPC system provides Internet access to business and residential customers via a "hybrid" network technology combining downstream satellite bandwidth and conventional upstream analog modem service. Downstream bandwidth is provided on an unused 6MHz television channel on a direct broadcast video satellite. Upstream bandwidth is provided via an ordinary telephone modem.

Figure 2.1 shows the typical DirecPC connection. The customer's computer—by convention called a "hybrid host" (HH)—forwards all outbound packets over the modem through an Internet Service Provider (ISP) to the DirecPC Network Operations Center (NOC). The heart of the NOC is the hybrid gateway (HGW). The hybrid gateway forwards packets to the chosen Internet server (IS). Packets comprising the server's response are received by the HGW, and forwarded back to the hybrid host over the satellite link.

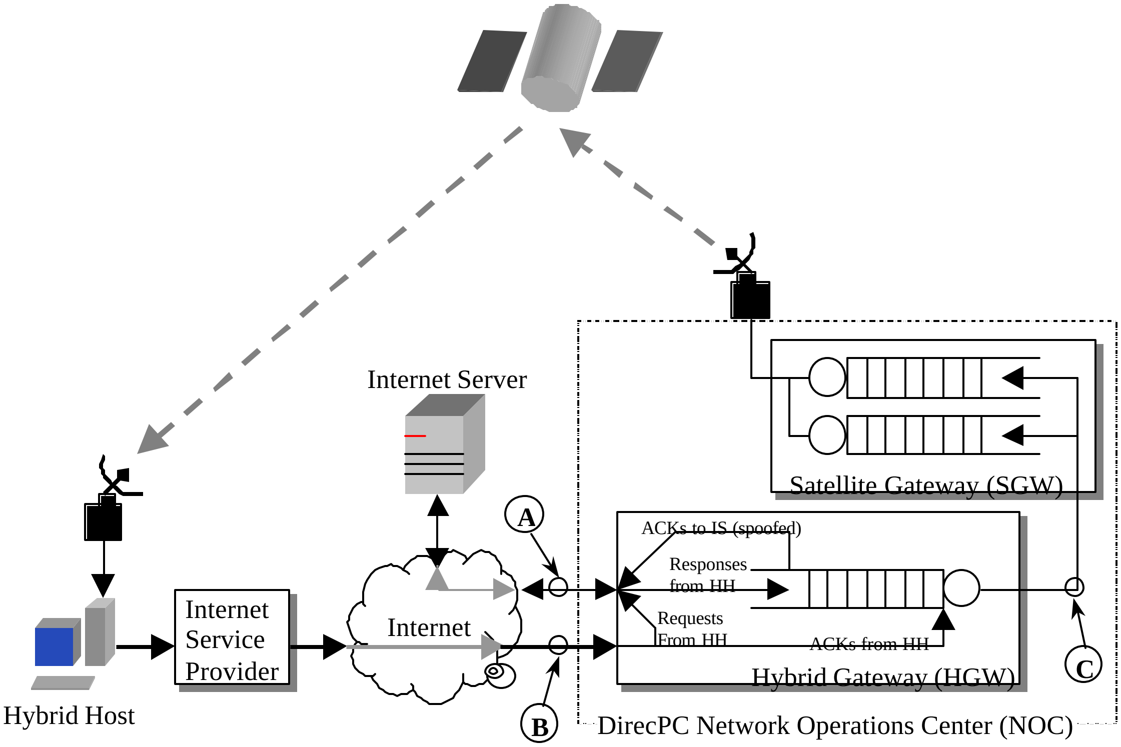

Figure 2.1 – Typical HSTN Configuration

Because the satellite sits in a geostationary orbit, the link delay from the HGW to the HH is approximately 250ms. The performance of conventional TCP over this long delay link will be hindered by its relatively small transmit window. The hybrid gateway addresses this problem through the use of connection-splitting. By acknowledging incoming packets from the IS on behalf of the hybrid host (thus assuming responsibility for their reliable delivery to the HH), and using the large windows TCP option (Internet RFC 1072 [JB88]) over the satellite link, the "apparent" round trip time experienced by the IS can be minimized. This results in the maximum achievable throughput. This approach is sometimes referred to as “spoofing” because this act of pretending to be the HH constitutes a benign form of IP address spoofing.

2.2 Technical Issues

This connection-splitting technique comes with a cost: considerable memory requirements on the HGW. When a packet is received from an IS it must be enqueued in a send buffer, to be forwarded to the HH. Once sent, a copy of the packet must be maintained in a retransmit buffer while awaiting acknowledgment by the HH. Typically the memory available for these buffers is divided equally among all connections, with caps on the total number of split connections and on the maximum amount of memory allocated to any connection.

The Satellite Gateway in the DirecPC system utilizes two priority queues for scheduling packets destined to various users. IP datagrams encapsulating any UDP packets, or TCP segments from connections that have not exceeded their buffer threshold in the HGW, are assigned to the higher priority queue. Those datagrams containing TCP segments belonging to connections which have exceeded their buffer threshold on the HGW are assigned to the lower priority queue. All datagrams in the higher priority queue are served before any in the lower priority queue.

2.3 Hybrid Internet Traffic Modeling

The connection-oriented nature of HSTN service affects traffic modeling for this environment. All of the popular self-similar time-series traffic models—such as Fractional Brownian Motion (FBM), Autoregressive, and Fractional Autoregressive Integrated Moving Average (FARIMA)—describe packet arrivals for aggregated traffic, but offer no framework for assigning individual packets to flows (or connections). These classes of models represent doubly-asymptotic models of network traffic; both in the sense of asymptotic for large (similar) numbers of users and in the sense of time asymptotic. However in the HSTN environment, we are interested in modeling traffic at the connection level (meaning characterizing connection traffic), because in any simulation model of the HSTN gateway (the HGW in the DirecPC scheme) we have to be able to input synthetic data (or real data) tagged by individual connections. Having fit an FBM or a FARIMA model to the aggregate traffic traces does not allow us to distinguish packets belonging to separate connections. Nevertheless, the traces we collected can allow us to study individual or aggregate connections' packets and their statistics and related models. Although these types of analyses were not undertaken in the present study, the data we have collected would allow for such studies in future efforts.

Since we can summarize each TCP connection by two characteristics: arrival time and the number of bytes transferred, we still have a broad class of models available to us. Assuming independence of arrival times and bytes transferred, we can fit any statistical distributions we please to these two variables. We will fit a total of 3 different distributions to both the interarrival times and connection sizes, though one of the connection size distributions will prove to be a very poor fit.

Chapter 3 – Network Traffic Models

From the earliest days of communication networks the development and application of accurate, readily-applied traffic models has been crucial to their successful deployment and growth. Accurate (high-fidelity) network traffic models are needed for planning and cost-effective dimensioning of network resources, and are the basis of quality of service guarantees. Traffic models have been a fundamental part of the success of modern telephone systems—allowing companies to provide a service that, in terms of reliability and cost, is (arguably) surpassed only by electric power. But the heterogeneity and complexity of wide area data networks and the applications generating their payloads have frustrated attempts to derive and apply simple traffic models for them. Much progress has been made recently in developing statistically accurate local-area and wide-area network traffic models, but their applicability has been hampered by their complexity. Furthermore, verification of their validity for large scale networks is still an open question.

3.1 Traffic History: The Poisson Model

Communication traffic theory has its roots in the engineering of early circuit switched telephone networks. Pioneering work by M.C. Rorty of AT&T modeled telephone calls as having a fixed length and calculated call blocking probabilities using a binomial expansion of the probabilities of the occurrence or non-occurrence of individual calls. E.C. Molina, also with AT&T, expanded this model when he independently derived a call arrival process that (it was soon discovered) had already been described by Siméon Poisson, whose name it came to bear. It was this model, completed by the work of A.K. Erlang of Denmark, who introduced the exponential holding time, that became the foundation of telephone traffic engineering [Fag75]. The exponential distribution, used for both the interarrival times and call holding times, takes the form:

$$F(x) = 1 - e^{-lx} \qquad (3\text{-}1)$$

with the corresponding density function:

$$f(x) = le^{-lx} \qquad (3\text{-}2)$$

where λ=E[x]$^{-1}$. A unique characteristic of the process is that the deviation is the same as the mean. A Poisson process is simply a random arrival process with exponentially distributed interarrival times. The most distinctive feature of this process is its randomness—it is completely memoryless. This model is elegant in that both the arrival process and the holding time distributions are each defined by only one parameter.

Synthetic Poisson data may be generated by generated the required number of samples, uniformly distributed between (0,1), i.e. u~U(0,1). Exponentially distributed sample values are then obtained through the transformation:

$$x=F^{-1}(u) \qquad (3\text{-}3)$$

where $F(\cdot)$ is the probability distribution function (in this case equation (3-1)).

3.2 Toward Modern Traffic Models

When computer networks first appeared, the Erlang model of Poisson call arrivals and exponentially distributed service times had ruled teletraffic for close to 50 years. Equations such as the Erlang "B" and "C" formulas, along with other mathematically tractable and elegantly derived results from queueing theory had lent themselves readily and reliably to network engineering. Moreover they were perceived as almost natural laws. So originally, and indeed for quite some time, despite the architectural and application changes, very little thought was given to re-evaluating the suitability of Poisson models to the traffic in the new networks that emerged in the last twenty years [WP98]. Call arrivals were simply replaced with packet arrivals, and holding times with "service" (or forwarding) times. Indeed, much of the early validation and peformance comparison work on Ethernet and ring networks—familiar papers by Metcalfe and Boggs [MB76], Bux [Bux81], and others—was based on this model. This is still the predominant model for packet traffic in network texts and university courses.

Over the years several extensions of the Poisson model have been suggested to improve its accuracy, including sums of multiple exponential distributions, and Markov-Modulated Poisson Processes (MMPPs) [Heff80]. Most have met with limited success, largely because they still relied on the exponential distribution. The MMPP model for instance has been successfully employed in the modeling of packetized voice and data traffic [HL86].

But it is now understood that the old rules do not apply to data networks. As cause Willinger and Paxson identify four significant ways in which data networks differ from voice networks [WP98]:

- data networks are packet based instead of circuit switched;
- individual connection durations and bandwidth requirements are variable;
- packets are buffered at points during transmission and may be dropped;
- most network layer protocols contain end-to-end congestion control mechanisms that introduce complex correlations.

3.3 Long Range Dependence

In 1990, investigating the types of traffic expected on future broadband networks, W. Leland and D. Wilson gathered the largest, most accurate interconnected LAN traffic trace of its time [LW91]. Its time-stamp precision and size gave a view of time-scales from milliseconds to months. The conclusion from this trace was indisputable: wide area traffic was bursty on much larger time scales than that provided for by Poisson-based modeling [FL91], and this long range dependence, or "self-similarity" (to be defined shortly) received much attention. In 1994 a slew of papers arrived finding evidence of self-similarity in Ethernet traffic [LTWW94], ISDN traffic [GW94], variable-bit-rate video traffic [BSTW94], Common Channel Signaling Networks [DMRW94], and Internet traffic [PF94].

The term "self-similarity" is a property of fractal processes, and in network traffic refers to a time scale characteristic: statistical similarity over a wide range of time-scale aggregations. That is, a continuous-time stochastic process $\mathbf{x}(t)$ is statistically self-similar with parameter H ($0.5 \leq H \leq 1$) if for any real $a > 0$, the process $a^{-H}\mathbf{x}(at)$ has the same statistical properties as $\mathbf{x}(t)$ [Stal98]. Likewise a discrete-time stochastic process is second-order self-similar if, for all m, the m-aggregated time series:

$$\mathbf{x}^{(m)} = \{\mathbf{x}_k^{(m)}, k = 0,1,2\ldots\} \qquad \text{(where } \mathbf{x}_k^{(m)} = \frac{1}{m}\sum_{i=km-(m-1)}^{km} \mathbf{x}_i \text{)} \qquad (3\text{-}4)$$

has the following variance and autocorrelation relationships with the original series $\mathbf{x}^{(m)}$:

$$\mathrm{Var}(\mathbf{x}^{(m)}) = \frac{\mathrm{Var}(\mathbf{x})}{m^{b}} \qquad \text{and} \qquad \mathrm{R}_{\mathbf{x}^{(m)}}(l) \to \mathrm{R}_{\mathbf{x}}(l) \text{ as } m \to \infty \qquad (3\text{-}5, 6)$$

The “Hurst” parameter, H, is a measure of the degree of self-similarity—or in other words, how well the statistical properties scale with respect to time. A value of 0.5 indicates no self-similarity, and 1.0 perfect self-similarity. The parameter β is the corresponding measure of self-similarity in the discrete time definition, and is related to H as β=2(1-H) [LTWW94].

Long-range dependence, a related phenomenon, is a statistical property of self-similar processes, and refers to a hyperbolically decaying autocovariance. Short-range dependent processes such as the Poisson process have (much faster) exponentially decaying autocovariances. The more slowly decaying autocovariances of self-similar processes reflect the persistence of their burstiness through many time scales [Stal98].

The finding of self-similarity does not immediately lead to a traffic model, as it is only a statistical property (of an infinite class of models). But mathematicians have long known that self-similar processes arise from the presence of so-called “heavy-tailed” distributions in the system [Ma63]. A heavy-tailed distribution is one which matches the proportionality:

$$\mathrm{P}(T > t) \propto t^{-g} \qquad (0 < \gamma < 2) \qquad (3\text{-}7)$$

for $t \to \infty$ [HLF98]. Heavy-tailed distributions have infinite variance and, for $\gamma < 1$, infinite mean.

The Weibull and Pareto distributions (to be discussed later) are two heavy-tailed distributions which, when incorporated into traffic models, produce self-similar behavior. They can be used to model interarrival times or connection durations (message lengths)—or both. Willinger, Taqqu, Sherman and Wilson have proven that the superposition of many ON/OFF sources with strictly alternating ON- and OFF- periods, both of heavy-tailed distribution, produces aggregate network traffic that is self-similar; they present results showing that it also closely matches Ethernet LAN traffic [WTSW97].

3.4 Log-normal Distribution

The log-normal distribution is sub-exponential, but does not have a strictly heavy-tailed probability density function. The definition of the log-normal distribution is based on the normal distribution, as follows: given that $Z = \log(X)$ is normally distributed (with zero mean), the random variable X shall be called log-normal. The log-normal probability density function takes the form:

$$f(x) = \frac{1}{x\sqrt{2p}\,s} e^{-\frac{(\log(x)-z)^2}{2s^2}} \qquad (x > 0) \qquad \text{(3-8)}$$

where z represents the mean, and s the standard deviation, of Z. These parameters can then be estimated as they would be for a normal distribution (given outcomes x_1, x_2, … x_n), by the maximum likelihood estimators:

$$\hat{z} = \frac{1}{n}\sum_{j=1}^{n}\log(x_j) \qquad \hat{s} = \left[\frac{1}{n}\sum_{j=1}^{n}(\log(x_j)-\hat{z})^2\right]^{\frac{1}{2}} \qquad \text{(3-9, 10)}$$

These estimators are unbiased [JK70]. The mean and standard deviation of the corresponding log-normal distribution X are thus:

$$E(x) = e^{z + \frac{s^2}{2}} \qquad s_x = e^z \sqrt{e^{s^2}(e^{s^2} - 1)} \qquad (3\text{-}11, 12)$$

The log-normal distribution is one of the earlier non-exponential distributions to be applied to network traffic modeling. The log-normal distribution has been suggested in the modeling of phone call durations [Bol94], local area network packet interarrivals [MM85], and Telnet connection sizes (in packets) and FTP data connection spacing [PF95]. We generate log-normal interarrival times and response sizes by transforming unit normal samples *u*~N(0,1) via the relation:

$$x = e^u \qquad (3\text{-}13)$$

3.5 Weibull Distribution

The Weibull distribution is a popular heavy-tailed distribution in network traffic modeling. The probability density function is:

$$f(x) = abx^{b-1}e^{-ax^b} \qquad (3\text{-}14)$$

and the distribution function is:

$$F(x) = 1 - e^{-ax^b} \qquad (3\text{-}15)$$

The maximum likelihood estimator for b is obtained by iteratively solving the equation [JK70]:

$$\hat{b} = \left[\left(\sum_{i=1}^{n} X_i^{\hat{b}} \log X_i\right) \Big/ \left(\sum_{i=1}^{n} X_i^{\hat{b}}\right) - \left(\frac{1}{n}\sum_{i=1}^{n} \log X_i\right)\right]^{-1} \qquad (3\text{-}16)$$

The estimator for the parameter a is then:

$$\hat{a} = \left[\frac{1}{n}\sum_{i=1}^{n} X_i^{\hat{b}}\right]^{1/\hat{b}} \qquad (3\text{-}17)$$

We generate Weibull trace values in the same manner as the exponential distribution, using the inverse of the Weibull distribution function (eq. 3-15).

3.6 Pareto Distribution

The Pareto distribution is a very popular heavy-tailed distribution, with density function:

$$f(x) = \frac{ak^a}{(x)^{k+1}} \tag{3-18}$$

(where k represents the minimum value) and distribution function:

$$F(x) = 1 - \left(\frac{k}{x}\right)^a \tag{3-19}$$

The variance is infinite if $\alpha \leq 2$, and the mean is infinite if $\alpha \leq 1$; otherwise the mean and standard deviation are as follows:

$$E[x] = \frac{ak}{a-1} \quad \text{and} \quad s_x = \sqrt{\frac{ak^2}{(a-1)^2(a-2)}} \tag{3-20, 21}$$

The parameter α is related to the Hurst parameter H as α=3-2H.

There are a number of ways to fit the Pareto distribution, including least squares, moments-based, maximum likelihood and iterative means [CM80], and Hurst parameter estimation via block packet count [HLF98]. We use the maximum likelihood estimator, which is as follows:

$$\hat{a} = \frac{n-1}{\sum_{i=1}^{n} \log X_i - n \log \hat{k}} \tag{3-22}$$

where $\hat{k}$ =min(X_i). This estimator is unbiased.

The Pareto distribution has been used to model the sizes of web pages, disk file sizes, and FTP-data bursts [PF95].

3.7 Fit Comparison

When fitting distributions to actual traces it is useful to have a discrepancy measure—particularly one that allows fit comparisons between different distributions. We use the l^2 test used by Paxson, Feldman and others [Pax94], [Feld95]. This test modifies the c^2 test to make its results independent of the number of bins used. To review, the c^2 test proceeds as follows. Given n outcomes of a random process X being fit to a model Z, we choose a partition of N equally spaced bins and define X_i as the number of outcomes in bin i. We further define p_i as the proportion of distribution Z falling in bin i. The discrepancy measure is then:

$$c^2 = \sum_{i=1}^{N} \frac{\left(X_i - np_i\right)^2}{np_i} \tag{3-23}$$

This measure is not independent of N, however, which presents problems when trying to compare results across different traces. A modification, suggested by Pederson and Johnson [PJ90], solves this problem. We first compute:

$$K = \sum_{i=1}^{N} \frac{\left(X_i - np_i\right)}{np_i} \tag{3-24}$$

followed by the "degrees of freedom":

$$df = N - 1 - Est \tag{3-25}$$

where Est is the number of parameters of Z being estimated from trace X. The new discrepancy measure $\hat{l}^2$ is then:

$$\hat{l}^2 = \frac{c^2 - K - df}{n - 1} \tag{3-26}$$

The standard deviation of $\hat{l}^2$ is:

$$s_{l} = \sqrt{\frac{2df + 4n\hat{l}^2 + 4n\hat{l}^4 + 4T}{n^2}} \tag{3-27}$$

where $$T = \sum_{i=1}^{N}\left[D_i^3 - 2D_iE_i + \frac{5}{2}D_i^2 + \frac{3}{2}\left(D_i + E_i\right)\right]\Big/E_i^2 \tag{3-28}$$

Thus far we have not discussed the bins, other than describing them as equally spaced. We use the following formula for bin width, according to [Pax94]:

$$w = 3.49\hat{s}_x n^{-1/3} \tag{3-29}$$

The number of bins, N, is then determined from w and the range of X. Because the number of outcomes will be very small in the bins near the tail, we follow [Feld95] and combine all bins with less than 5 outcomes.

Chapter 4 – Traffic Traces and Modeling

4.1 DirecPC™ Trace(s)

For this study three sample traces were taken of actual DirecPC NOC traffic, all using a modified version of the *tcpdump* program [Jac98]. All traces were taken using a Linux PC equipped with a 100 Base-T Ethernet adaptor and a high resolution (~10µs) timer.[1] The network vantage point for the trace logging was a spanned port on the primary NOC router, giving us access to all packets passing between hosts on the NOC LAN, as well as all packets inbound or outbound on the Internet links (two T-3s).

Referring back to Figure 2.1, we could have chosen to collect our trace at points A, B, or C (or really any combination of these, since each of these links represents a hope through the core router). The path between the HGW and the SGW (point C) is a virtual LAN, and we excluded it from the port span, so our traces contain only the packets seen at points A and B. This means that each packet inbound from a HH was actually logged twice: once in its tunneled form passing through point A to the HGW, and once in its

[1] The default timer granularity of tcpdump is only about 10ms, but patches are widely available to improve it.

normal form passing from the HGW to the Internet. For this study we ignored the tunneled packets (point A) but this data was included in the trace to allow future studies of HH request patterns, etc.

The traces contain the first 100 bytes—beginning with the Ethernet header—of each Ethernet frame observed on the network. Together they contain over 75 million packets. In each case, at trace termination *tcpdump* reported no dropped packets, so the traces can be considered complete. Each trace was postprocessed using the *tcp-reduce* script [Pax95] to produce one line summaries of each connection. Connections which failed to complete the setup phase were discounted, as well as all connections involved in updating the cache of DirecPC's Cacheflow caching appliance.

4.1.1 Trace Summaries

Trace one was taken on May 12, 1999, between the hours of 5 and 6pm Eastern Time. It is 3,473 seconds (0h:57m:53s) long and contains 86,062 complete TCP connections. Trace two was taken on October 13, 1999 between the hours of 5 and 6pm Eastern Time. It is 3,688 seconds (1h:01m:28s) long and contains 64,553 complete TCP connections. Trace three was also taken on October 13, 1999 during the peak hours of 10:30pm - 12:30am Eastern Time. It is 7,396 seconds (2h:03m:16s) long and contains 140,667 complete TCP connections. These traces are summarized in Table 4.1. The number of bytes shown reflects only the downstream direction, because it is this direction we are concerned with.

Trace	WWW	FTP-ctrl	FTP-data	POP	NNTP	Other	Total	(Units)
1	73,205	386	648	5,271	130	6,422	86,062	Connections
	852,949,541	227,390	406,206,429	54,921,840	109,907,689	97,960,411	1,522,173,300	Bytes
	85.06 %	0.45	0.75	6.12 %	0.15 %	7.46 %	100.00 %	Connections
	56.03 %	0.01	26.69	3.61 %	7.22 %	6.44 %	100.00 %	Bytes
2	53,754	310	601	3,661	133	6,094	64,553	Connections
	593,022,253	209,950	170,624,961	56,103,608	314,068,168	323,321,547	1,457,350,487	Bytes
	83.27 %	0.48 %	0.93 %	5.67 %	0.21 %	9.44 %	100.00 %	Connections
	40.69 %	0.01 %	11.71 %	3.85 %	21.55 %	22.19 %	100.00 %	Bytes
3	123,561	1,708	1,719	4,663	551	8,465	140,667	Connections
	1,711,701,945	988,441	1,368,561,512	52,761,724	923,483,754	576,591,034	4,634,088,410	Bytes
	87.84 %	1.21 %	1.22 %	3.31 %	0.39 %	6.02 %	100.00	Connections
	36.94 %	0.02 %	29.53 %	1.14 %	19.93 %	12.44 %	100.00	Bytes

Table 4.1 – Summary of traffic traces used

Upon examining the traces several things are evident. The first is that the bulk of the traffic is generated by two applications: the World Wide Web and File Transfer Protocol. Second, there is no Telnet or Rlogin traffic; which is because those low bandwidth, fast response demanding applications are routed back over the telephone network, to avoid incurring the satellite delay. Also notable is that FTP-data connections, while few in number, are very large in size. The same is true for NNTP. And most importantly, the significant majority of the traffic is of varieties that can be described as single-burst retrievals. That is, immediately upon connection-open a single object is downloaded (or perhaps a group of associated objects in immediate succession—i.e. e-mail messages in the POP case) and the connection closes shortly thereafter. Only the FTP-control, NNTP and Other[2] traffic varieties—comprising only about 10% of connections and 30% of bytes—may contain periods of both bursts and lulls in downstream traffic. And for the largest NNTP connections the bursts vastly exceed the

[2] In the case of trace 2 the unusually high amount of "Other" traffic can be attributed to one user who was downloading multiple 15 megabyte files via the "IRC" application—analogous to FTP and also "single-burst".

lulls. Hence we can conclude that the bulk of our traffic can be described as "single-burst" downloads.

4.1.2 Trace Validation

An assumption used in fitting traffic models to these traces is the statistical independence of connection arrival times and download sizes. Autocorrelation plots for all three traces, shown in Figure 4.1, reveal an adequate, though not particularly high, degree of independence in the connection arrival times. For traces 1 and 2 $r(1)$ is approximately 0.1, and for trace three it is closer to 0.15. While these are small enough to label our independence assumption valid, we shall pause to note here that they do indicate the potential for success using connection level time-series models, were such models available.

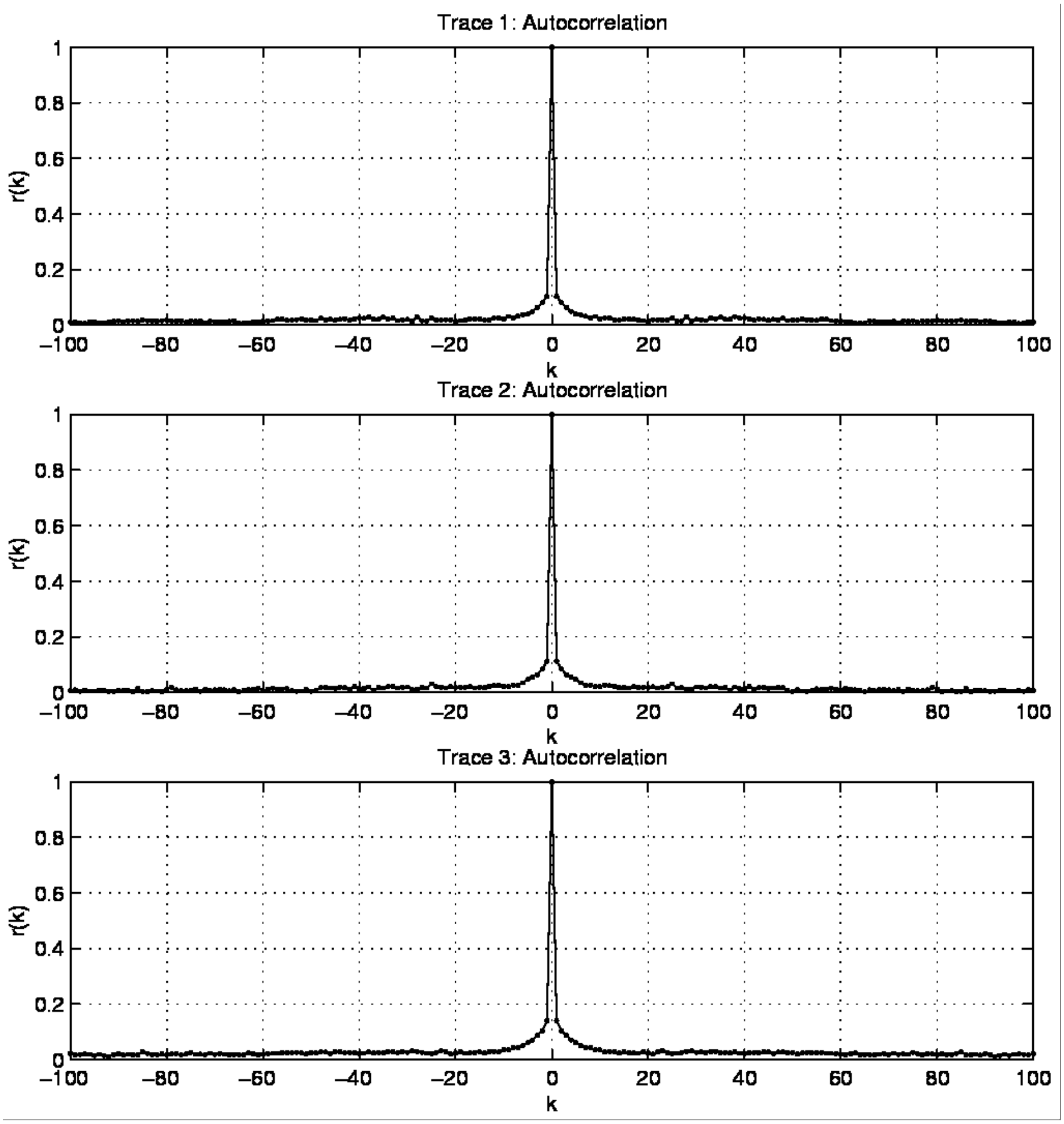

Figure 4.1 – Connection Interarrival Autocorrelations (for traces 1,2,3)

For the download sizes there is no question. Autocorrelation plots, shown in Figure 4.2, indicate a high degree of independence for this variable.

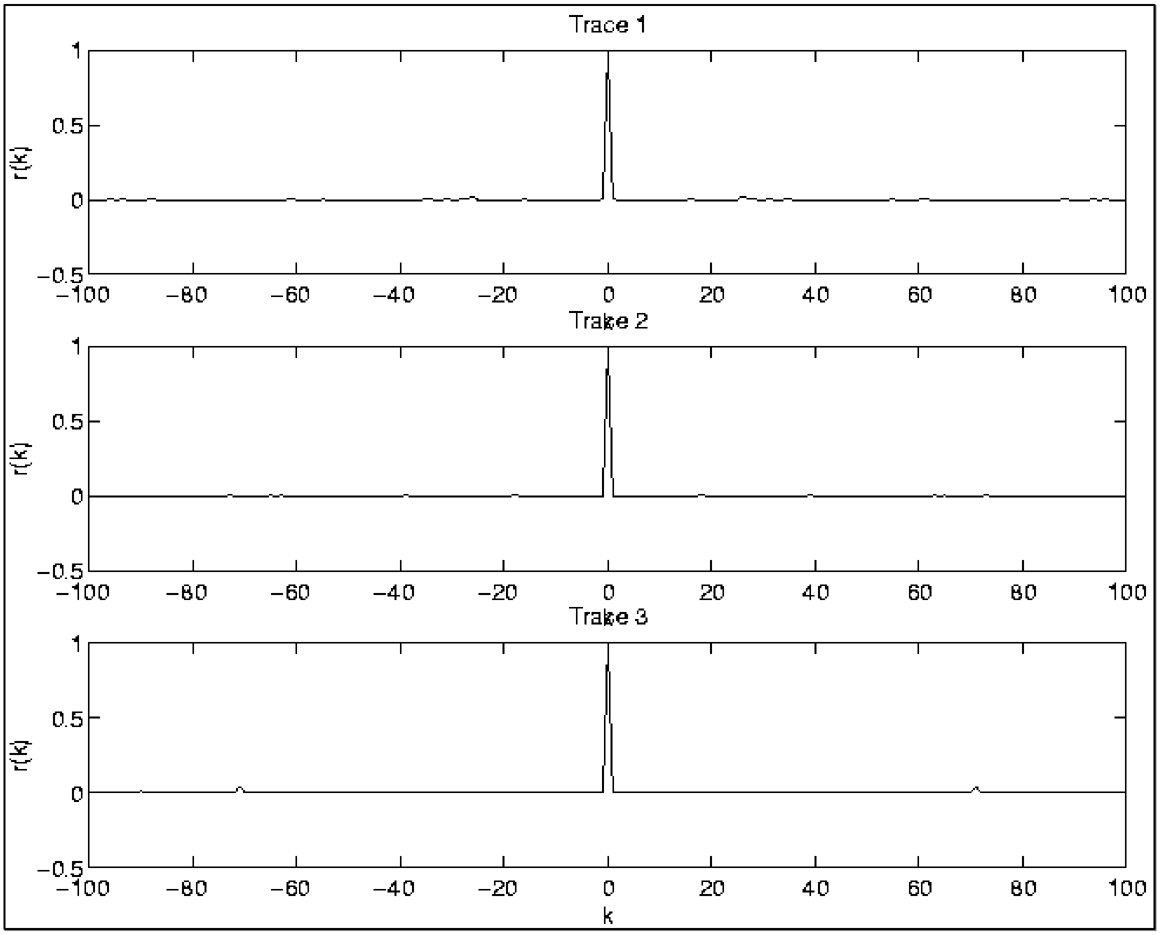

Figure 4.2 - Download Size Autocorrelations (for traces 1,2,3)

4.2 Connection Interarrivals

Connection interarrival distributions for traces 1, 2 and 3 are shown on a linear plot (highlighting the lower tails of the distributions) in Figure 4.3.

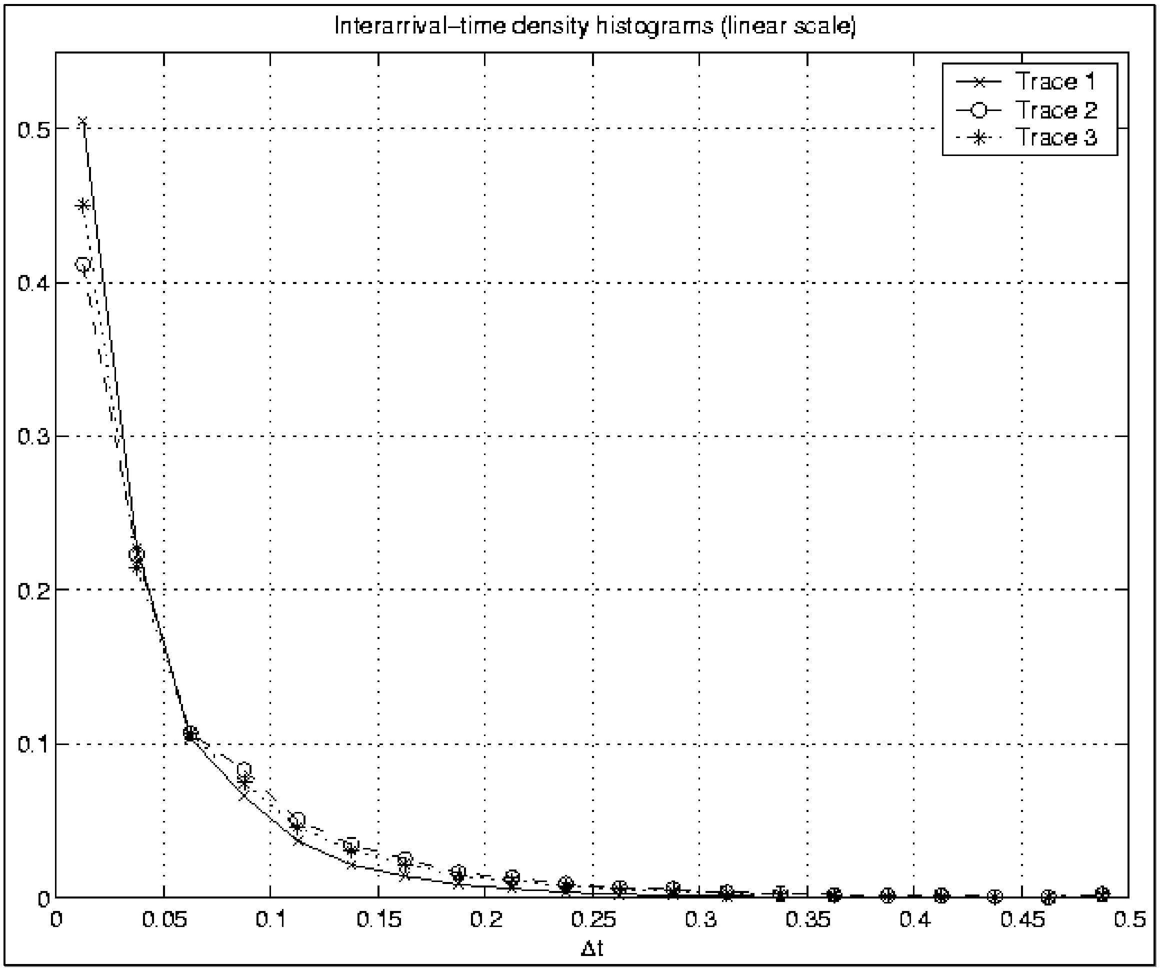

Figure 4.3 – Distributions of interarrivals (linear scale) (traces 1,2,3)

To better emphasize the tails, Figure 4.4 shows the same data on a semilog plot.

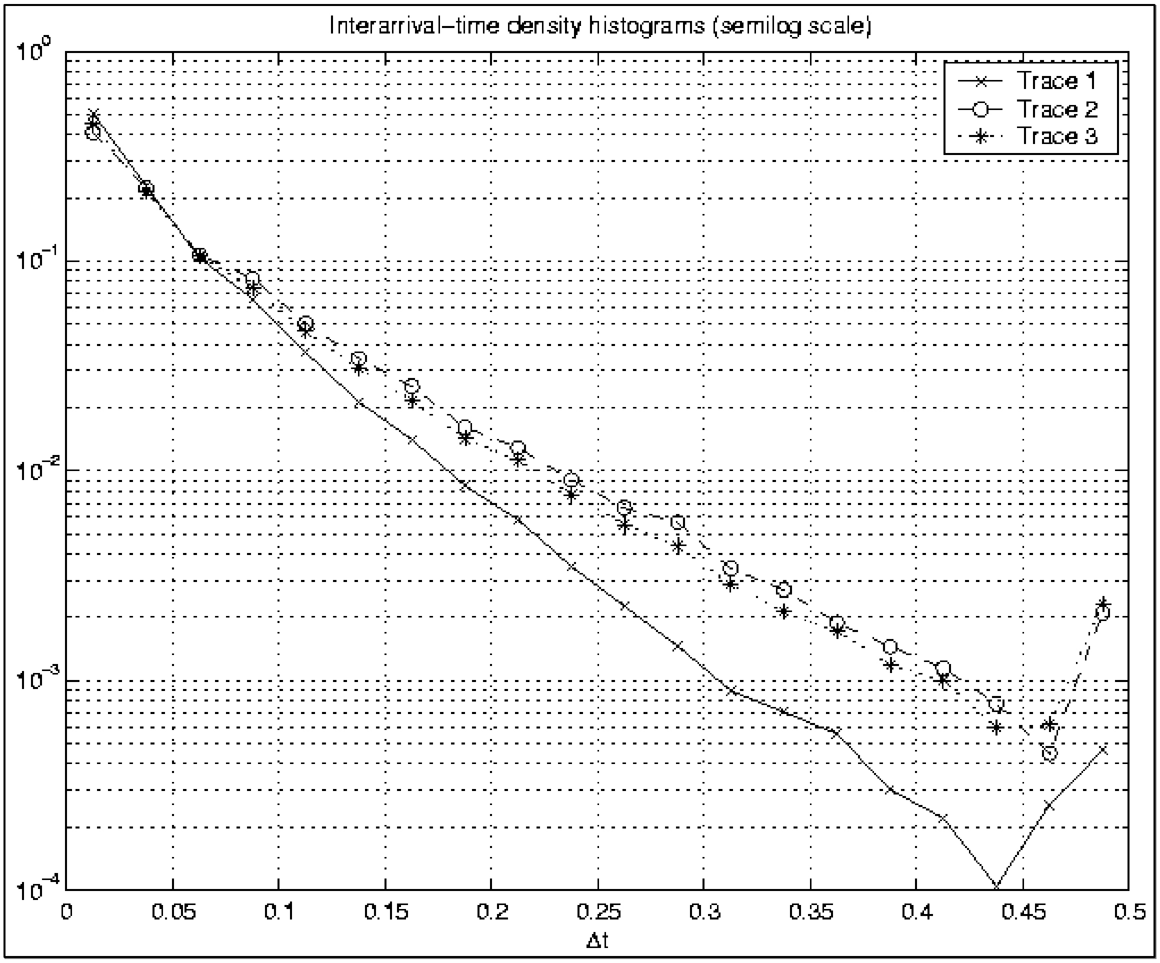

Figure 4.4 – Distributions of interarrivals[3] (semilog scale) (traces 1,2,3)

Fitted exponential, log-normal and Weibull distributions for traces 1, 2 and 3 are shown in Figures 4.5, 4.6, and 4.7, respectively. All three traces' interarrivals are plainly subexponential; the fitted exponential distributions obviously fail to capture their tail behavior. The log-normal distribution, on the other hand, is too heavily-tailed, and also appears to be a poor fit to all three traces. Fortunately the Weibull distribution, in all three cases, appears to be a good match.

[3] In this and all distribution plots in this work the last data point reflects the sum of the remaining tail.

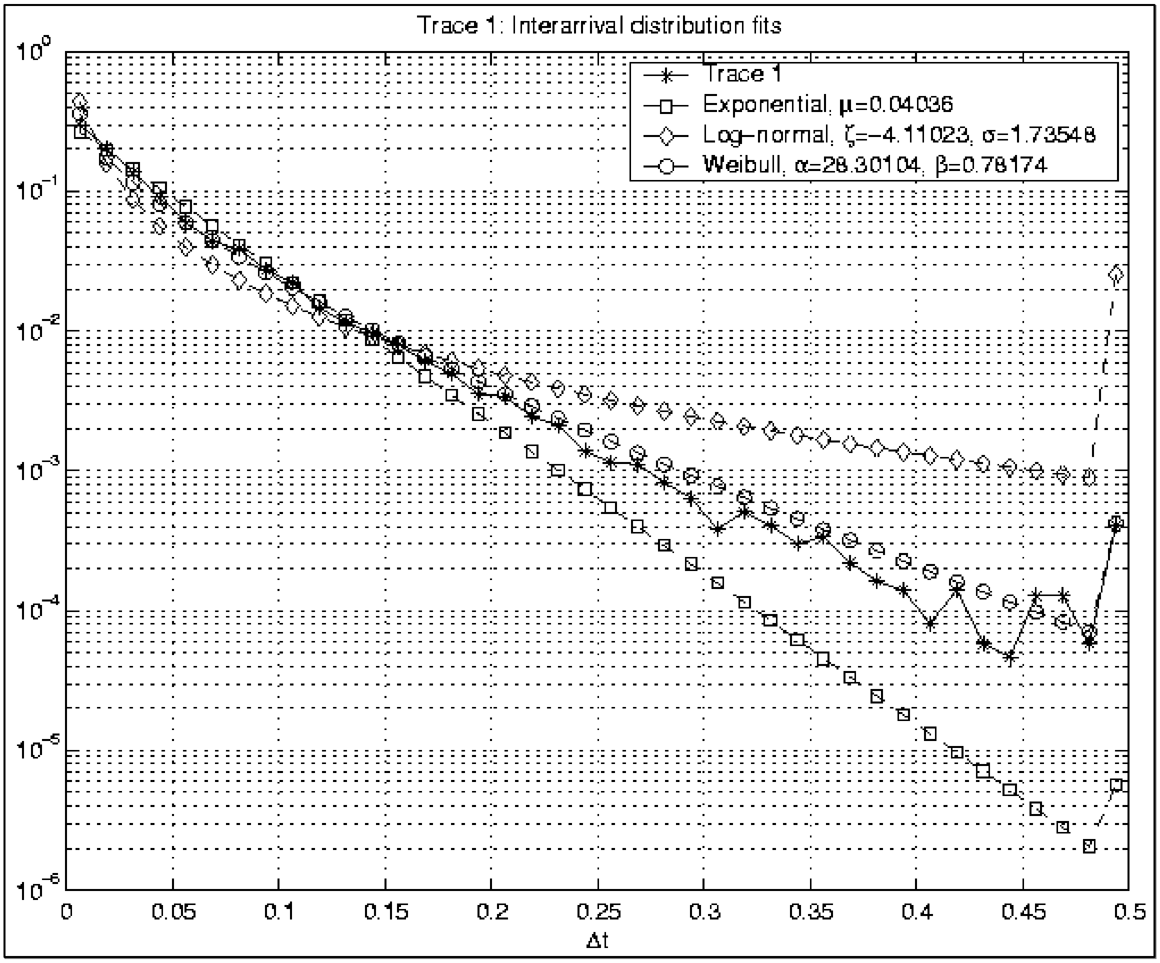

Figure 4.5 - Interarrival distribution of Trace 1

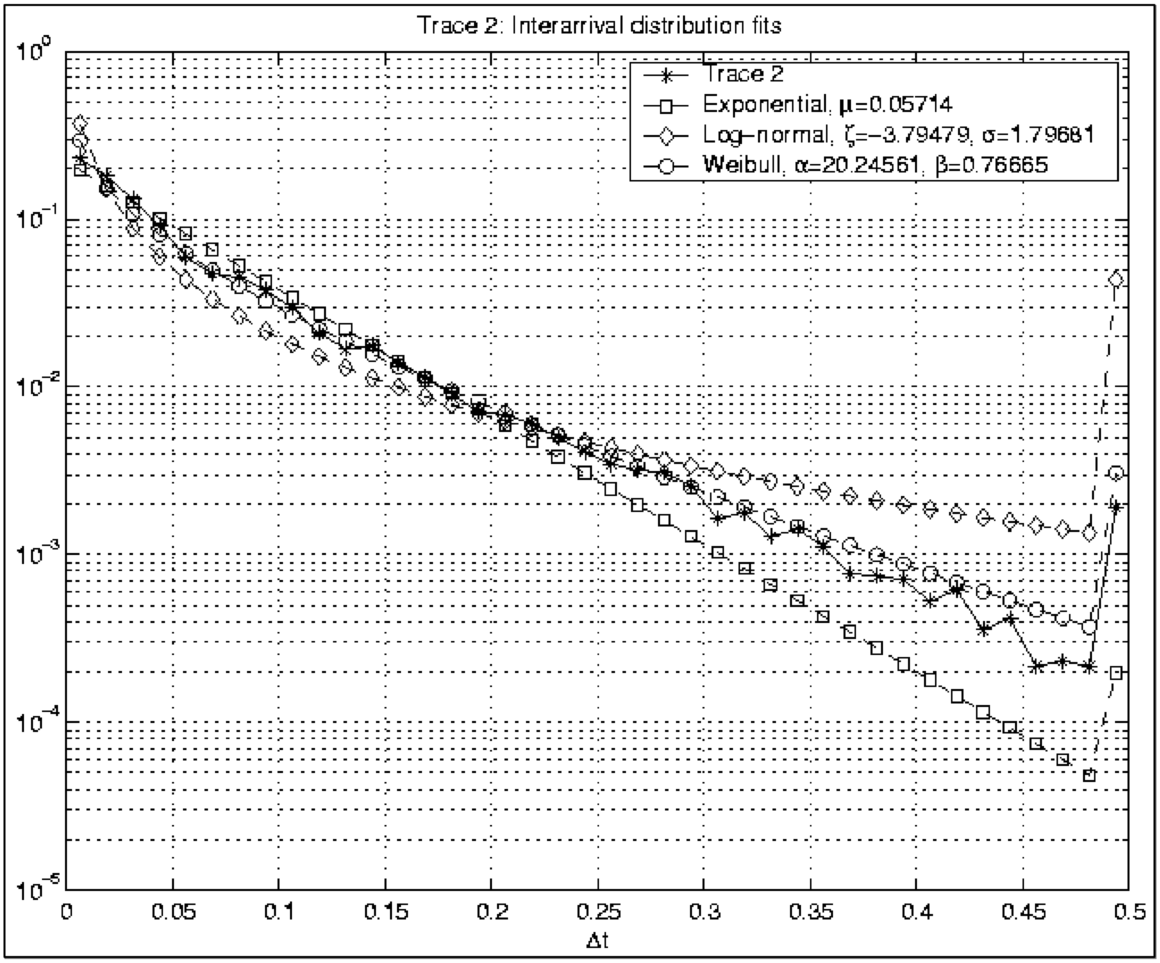

Figure 4.6 - Interarrival distribution of Trace 2

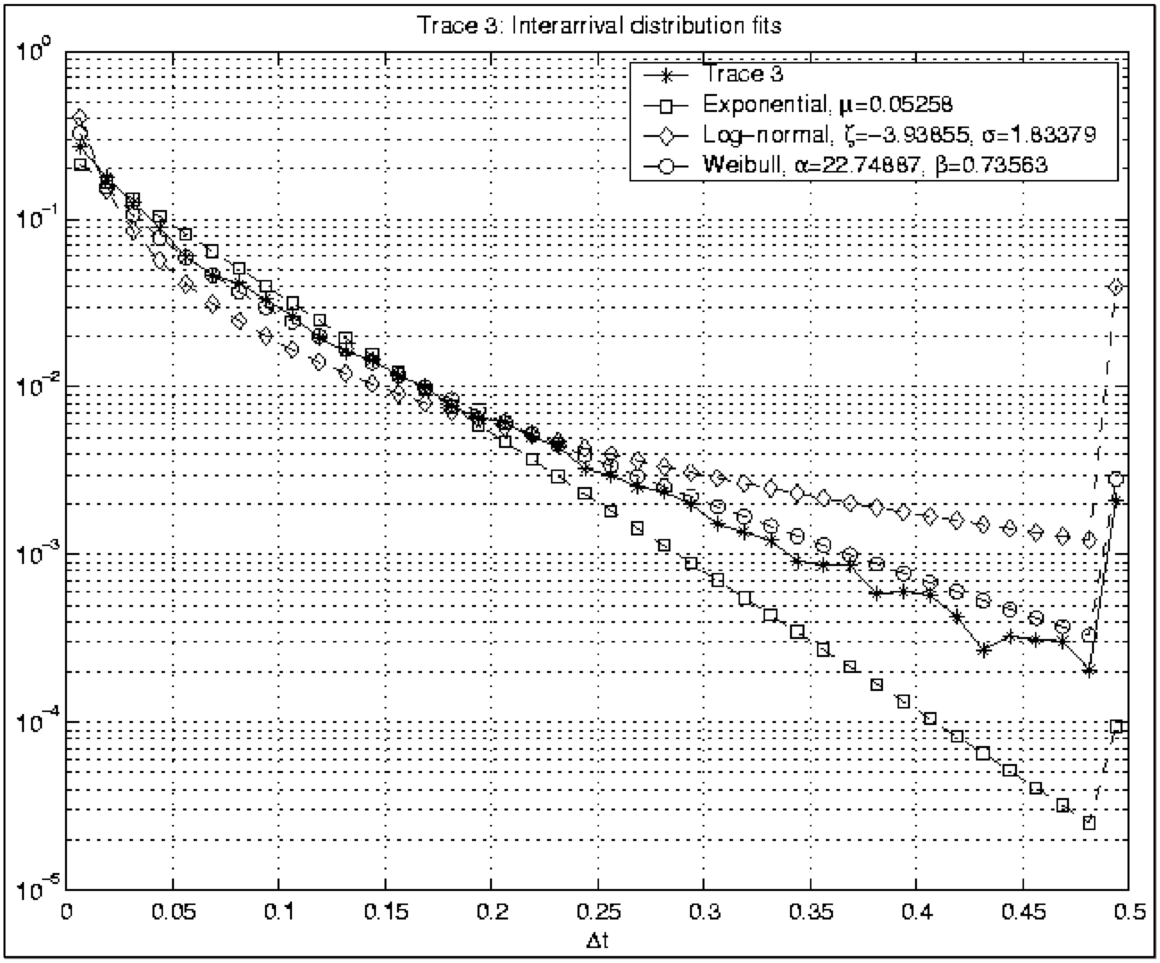

Figure 4.7 - Interarrival distribution of Trace 3

The suitability of the Weibull distribution is attested to by the results of the λ^2 test, shown in Table 4.2. For all three traces the Weibull distribution outfits the exponential and log-normal by a measurable margin. The high value of $\hat{l}^2$ for the exponential fit to trace 3 is not an error. Rather, it is proof of the substantially sub-exponential tail of the connection interarrivals. The largest arrival interval contained in trace 3 is 2.54 seconds. The expected number of exponential arrivals (np_i in equations (3-23) and (3-24)) in the bin containing this value is on the order of 10^{-12}, which contributes greatly to the size of $\hat{l}^2$.

Trace	Model	$\hat{l}^2$	$\hat{s}$
1	Exponential	2.037116e-01	1.551436e-02
	Log-normal	2.485222e-01	3.687781e-03
	Weibull	8.229092e-02	2.038071e-03
2	Exponential	2.823670e-01	8.147982e-02
	Log-normal	2.734092e-01	4.538923e-03
	Weibull	1.014115e-01	2.668935e-03
3	Exponential	3.424973e+02	3.992822e+02
	Log-normal	2.383435e-01	2.829614e-03
	Weibull	7.549240e-02	1.592505e-03

Table 4.2 - Interarrival l^2 goodness of fit results

4.3 Bytes downloaded (per connection)

Connection download sizes are considerably heavy-tailed. Figure 4.8 shows the actual response size distributions of all three traces, plotted against exponential distributions fit to each one. The tails are enormous: for all three traces the upper 5% of connections account for over 80% of all bytes downloaded. For this reason we immediately eliminate the exponential distribution as a candidate, and focus on fitting the three sub-exponential distributions to the download sizes.

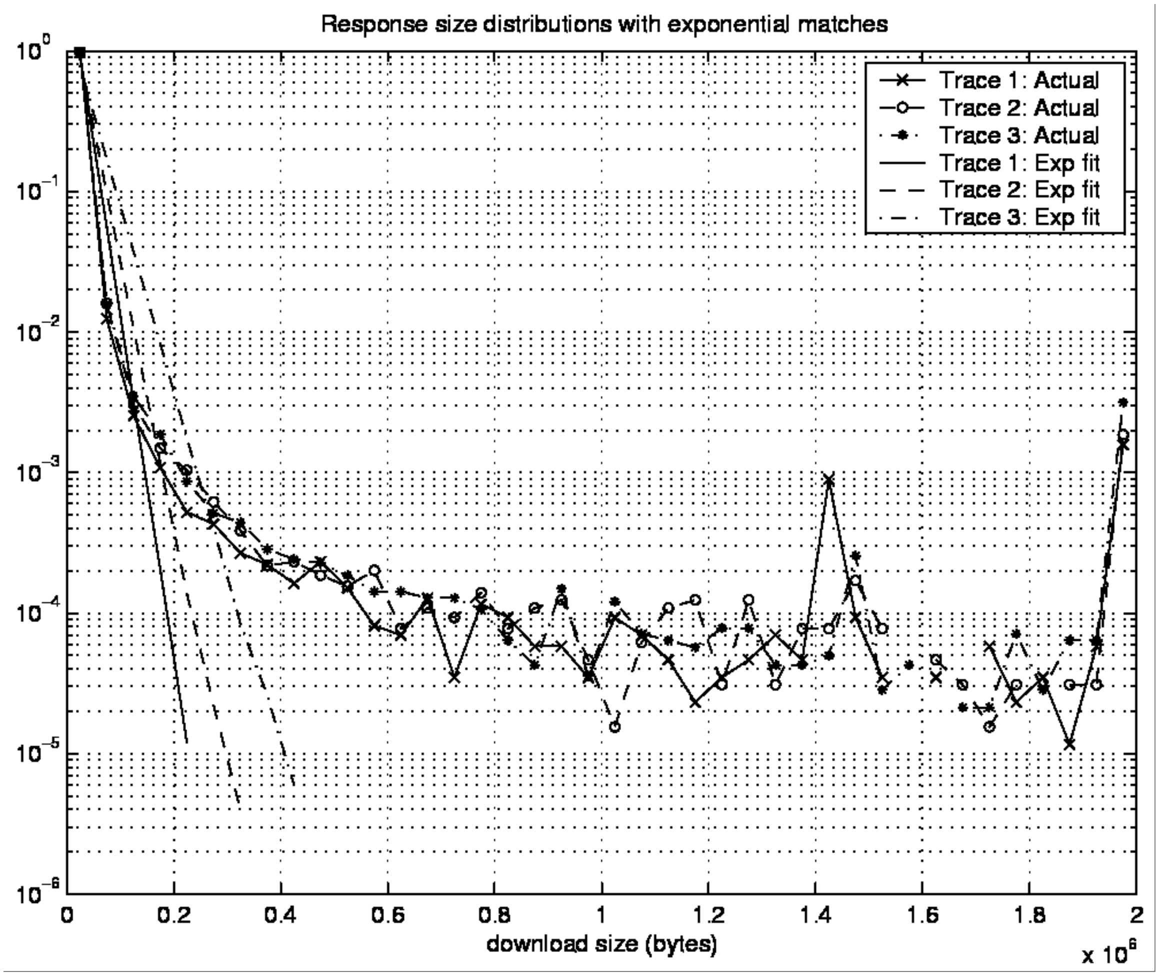

Figure 4.8 - Download size distributions

Figure 4.9 shows fitted log-normal, Weibull and Pareto distributions for the download sizes in traces 1. Table 4.3 shows the $\hat{l}^2$ best fit results for the same fits. There is a noticeable problem. Though the conclusion that the log-normal distribution is the best fit appears to be correct, the fit results for the Weibull and Pareto distributions are misleading. The tail on the Weibull distribution is so large, in fact, that its mean is over 10^6 bytes, though the mean download size of actual trace is approximately 1.7×10^4. The results for traces 2 and 3 proved to be similarly disastrous. Our conclusion is that the $\hat{l}^2$ goodness-of-fit test, while very useful for comparing fits to the moderately heavy-tailed interarrival data, is misleading when dealing with such heavily-tailed data as the download sizes. It appears to penalize a distribution much more for underestimating the upper tail than for overestimating it.

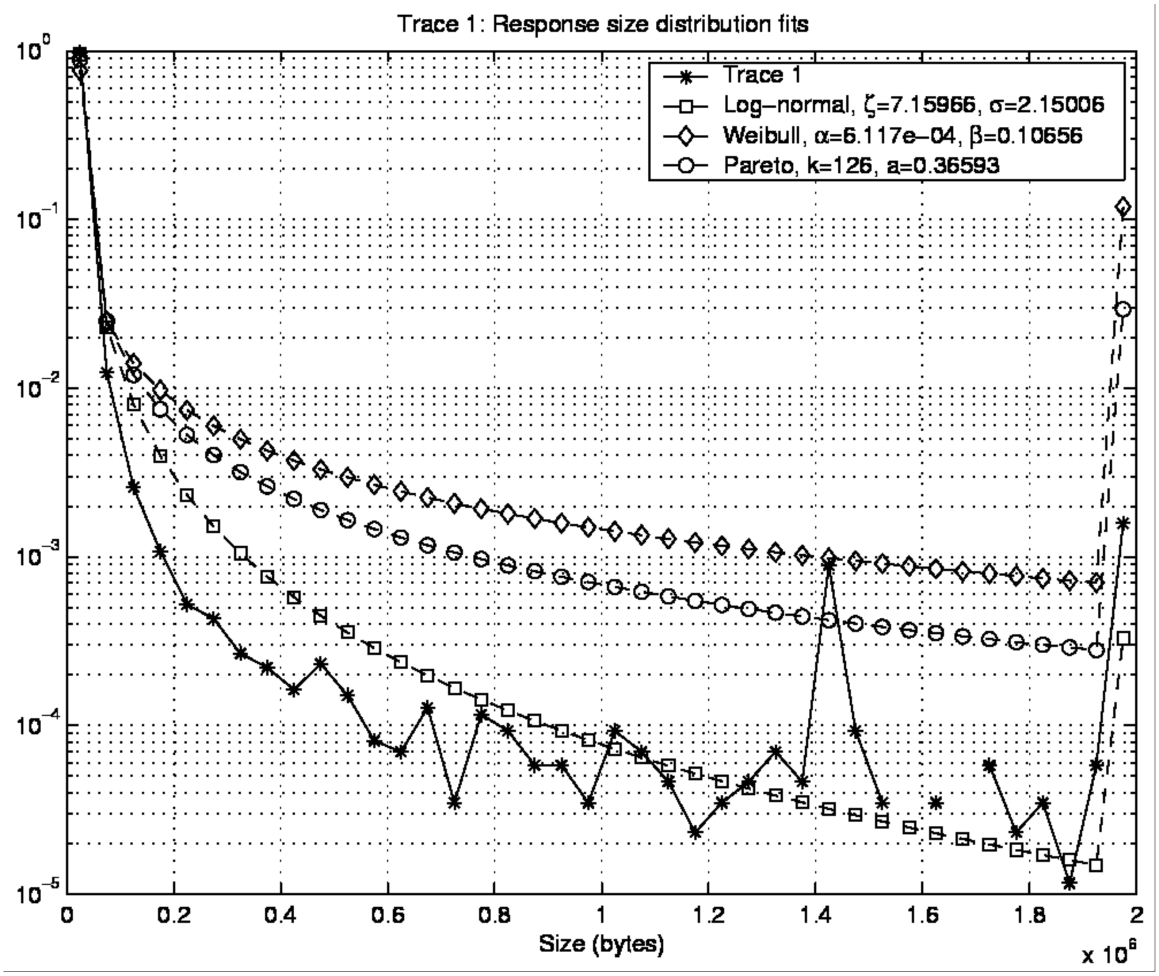

Figure 4.9 - Trace 1 download sizes (bytes)

Model	$\hat{l}^2$	$\hat{s}$
Log-normal	7.572985e-02	1.049059e-02
Weibull	2.095553e-01	2.410206e-03
Pareto	7.914014e-02	1.444359e-03

Table 4.3 – Download size l^2 goodness of fit results for trace 1

There is a similar problem for the Pareto distribution. The Pareto distribution is unique among the three because it has a non-zero minimum value. Historically the Pareto distribution has been fit to the upper tails of statistical data, rather than entire distributions. Here we must choose the best way to apply the Pareto. Consequently, we must fit the Pareto to an upper fraction of the entire data, but use it to model the entire distribution. Experiments indicate that fitting the Pareto to the upper 90% of the distribution gives the best "goodness-of-fit" results when evaluated against the entire data set. For our traces this led to a k of about 130 bytes. This is what is shown in Figure 4.9. However fitting the Pareto in this manner results in the same problem seen with the Weibull—overestimation of the upper tail. Hence, for our work we will fit the Pareto to the upper 20% of the download size data. This leads to a k of roughly half the mean response size (8-10kbytes).

Figures 4.10, 4.11, and 4.12 show the distribution fits (using the new 20% Pareto) for traces 1 through 3. For all three traces the log-normal distribution appears to be the best fit, but the Pareto now appears to be a more viable alternative. The Weibull is again shown, but only for comparison purposes. With sufficient evidence of its lack of fit to download sizes, we shall eliminate it from further consideration to model this traffic variable.

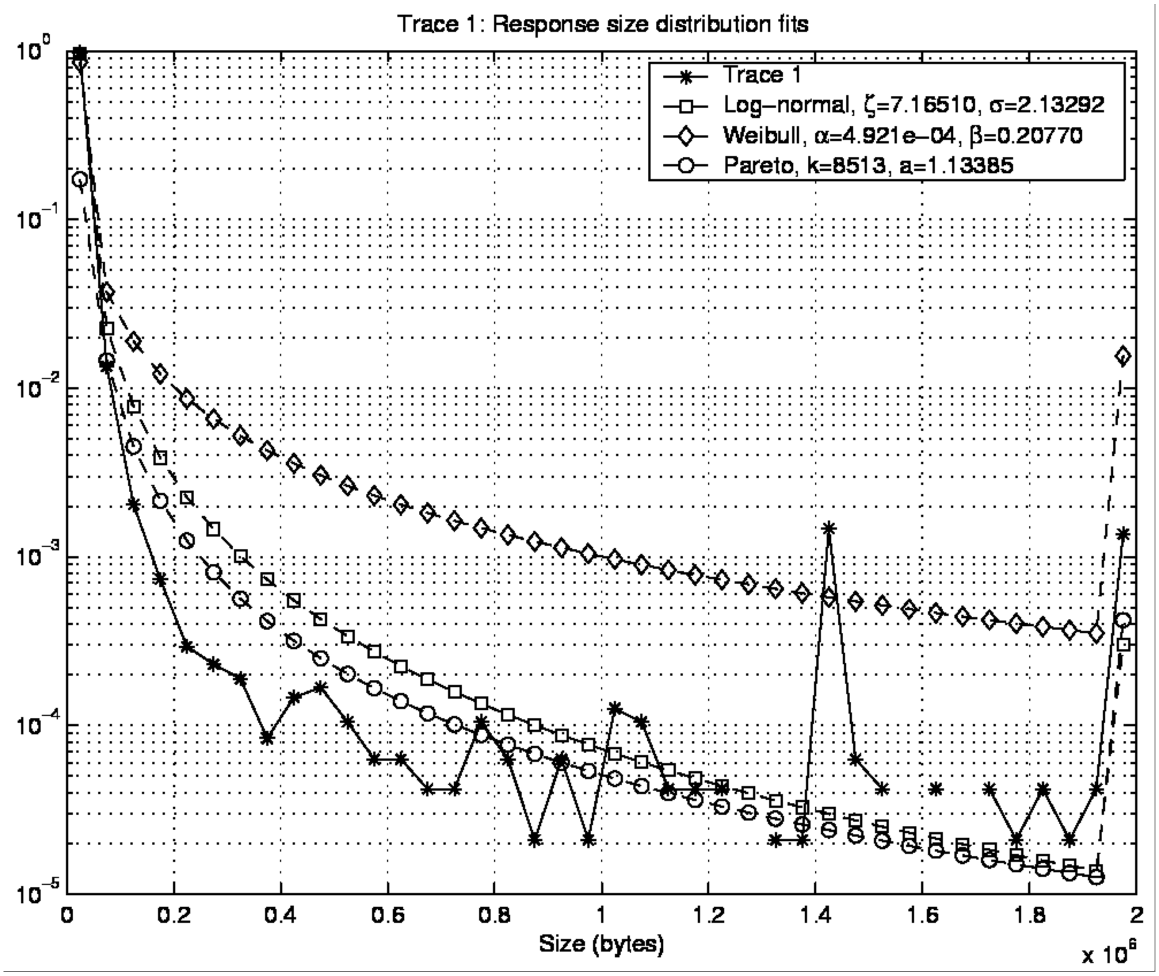

Figure 4.10 - Download size fits for trace 1

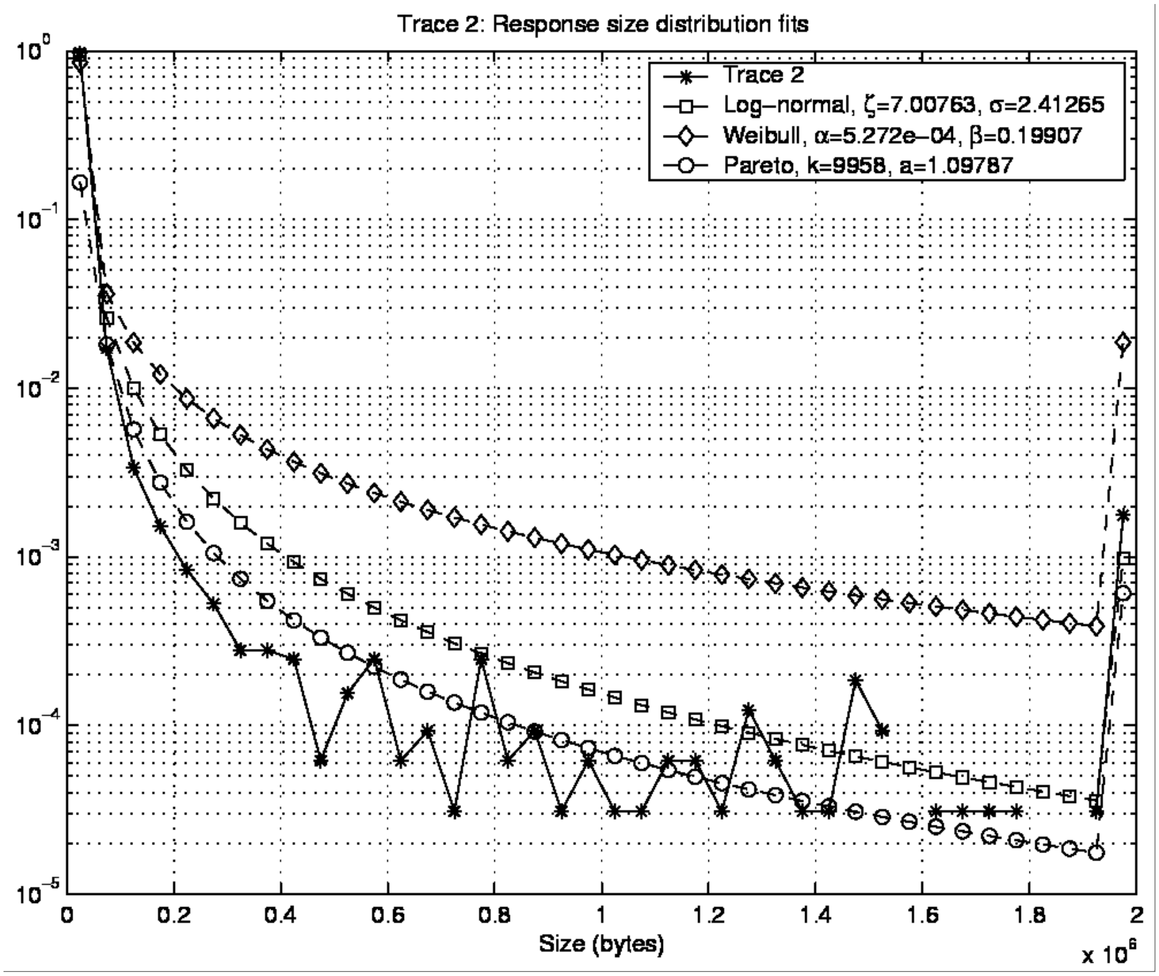

Figure 4.11 - Download size fits for trace 2

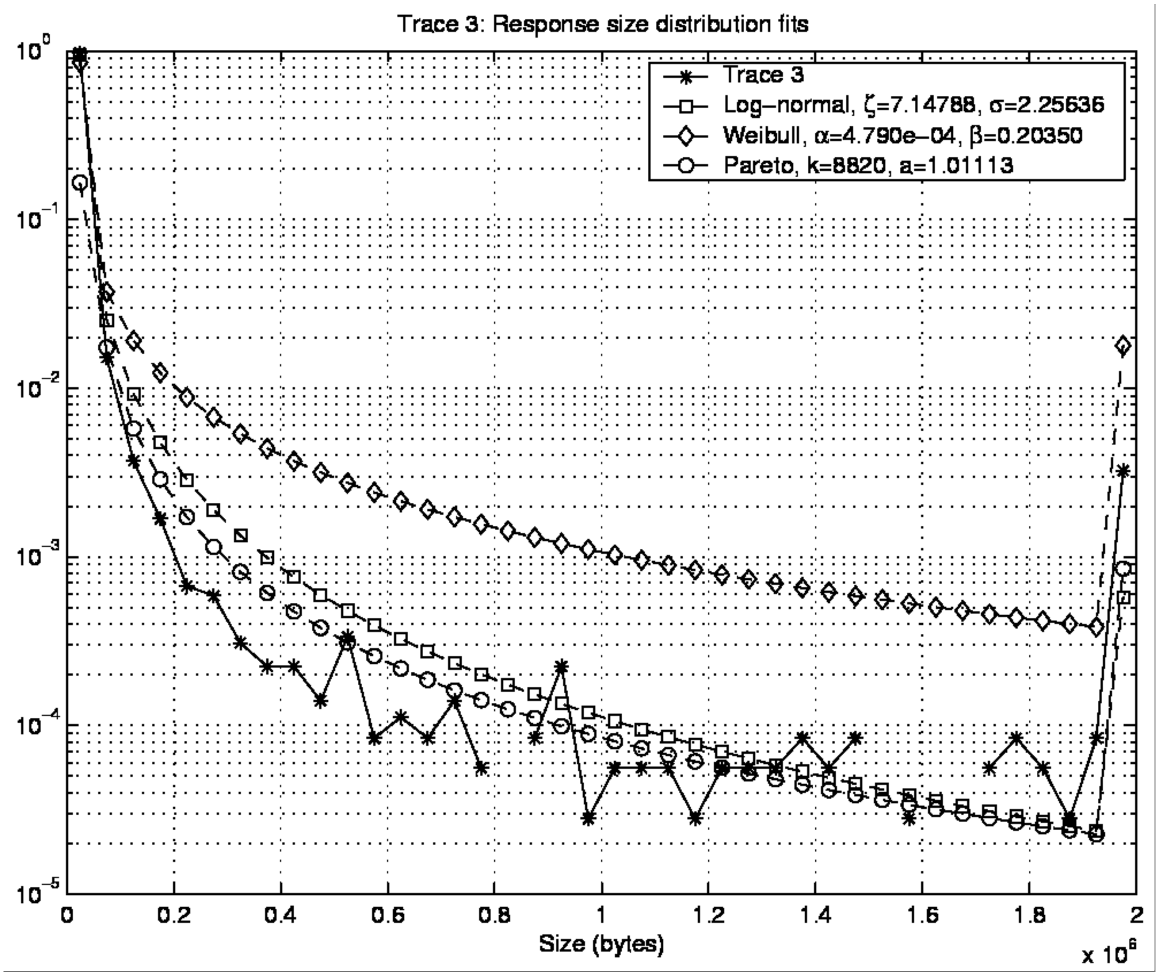

Figure 4.12 - Download size fits for trace 3

4.4 Summary

In short, we have found the following: DirecPC HSTN traffic can be modeled at the connection level by fitting statistical distributions to two key traffic variables: TCP connection interarrival times, and downstream transmission (“download”) sizes. TCP connection interarrival times are neither truly heavy-tailed, nor truly exponential, but fall somewhere in between, under the general classification of subexponential. Exponential, log-normal and Weibull are all reasonable candidate distributions for the interarrival times, with Weibull appearing to be the best fit.

TCP download sizes are extremely heavy-tailed, and rather difficult to fit. Standard goodness-of-fit tests are not only useless, but even misleading for this variable. With care, however, and using the approach we have show, a good fit can be obtained for this variable with the log-normal and Pareto distributions.

Also of note is the fact that, since the log-normal distribution is the best fit to the download sizes, they do not have infinite variance. This has important implications for our modeling, implying that the prediction task should not be as difficult.

Chapter 5 – HSTN Testbed Simulations

5.1 Opnet HSTN Testbed

For the purpose of studying the utility of different traffic models in the HSTN setting, a flexible HSTN-like environment was needed. An HSTN simulation testbed was constructed using the OPNET discrete event simulator. The OPNET environment provides complete, validated TCP/IP models, along with an isolated, fully configurable setting in which to run simulations. This yields results that are both realistic and repeatable—which is necessary for the traffic model comparison undertaken.

The testbed was constructed as a general model of an HSTN network. It is based on the DirecPC scheme, but with some secondary behaviors ignored, and a few parameters generalized.

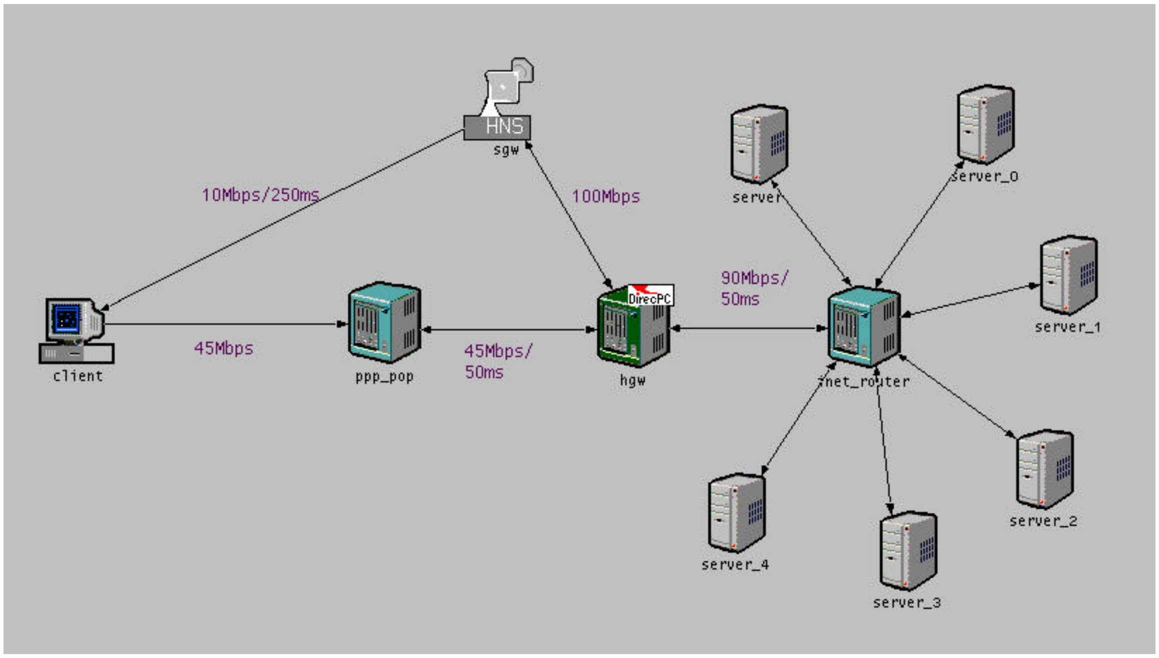

Figure 5.1 – Opnet HSTN Testbed Network

The testbed itself, shown in Figure 5.1, closely resembles a simple HSTN network, containing one of each of the fundamental pieces of the system presented in Figure 2.1. There is a client (HH), which sends outbound (request) packets to the hybrid gateway (HGW) via a dialup PPP connection; six servers, which return responses to the requests forwarded by the HGW; and a satellite gateway (SGW), through which the hybrid gateway forwards the response packets back to the HH. All interconnecting links are set to typical values, with the satellite link replaced with an equivalent point-to-point link (250ms delay), for simplicity. Intermediate routers are eliminated because their effect on the system is secondary.

5.1.1 Testbed Hybrid Gateway

In modeling the DirecPC Hybrid Gateway (HGW) there are several significant functions which must incorporated to accurately model the HGW behavior. First, the model must perform connection splitting/spoofing. Secondly, it must manage the number of active connections, and the memory available to each for use as a retransmit buffer. Thirdly it must advertise (to the sender) a receive window which is in compliance with both the receiver's advertised window size, and its own available buffer space. Fourthly, it must assign “priorities” to connections, based on whether or not they are overusing their available buffer, and embed these priority tags for the Satellite Gateway (SGW) to use.

The hybrid gateway (HGW) model is a based on an earlier model of a spoofing gateway written at the Center for Satellite and Hybrid Communication Networks (CSHCN) at the University of Maryland and used in other Internet over satellite and HSTN studies [KLBB99] [LKRB99]. It is built on a basic router model, but with extensive additions and modifications, including the addition of a TCP layer capable of spoofing/connection splitting. The IP layer is modified to examine datagram contents, and forward all TCP segments up to the TCP layer, which is modified to spoof acknowledgements and split connections. This earlier basic spoofing model was extended with additional functionality to duplicate all of the essential behaviors of the DirecPC hybrid gateway, including all those enumerated above.

5.1.2 Testbed Satellite Gateway

The satellite gateway model is essentially a modified router, possessing only an IP, and no TCP layer. As in the DirecPC scheme it is a two priority queueing system. To determine an incoming packet’s priority, the SGW examines the TCP header for a priority assignment given by the HGW. If the packet is not TCP or contains no priority tag, it will be assigned to the higher priority queue. Only when the higher priority queue is empty is the lower priority queue served. By default the SGW is configured with infinite queues, but is also switchable to finite queue lengths (to study drop probabilities).

5.1.3 Testbed Client/Server

Key to studying traffic models is incorporating traffic traces, whether previously captured or synthetically generated, into the network. This was accomplished in the testbed by a significant modification of the Generalized Network Application (GNA) client and server models found in OPNET. The default GNA models generate common TCP network traffic like HTTP, FTP and other traffic, but provide little flexibility for incorporating other traffic models or statistical distributions. A "trace file" application was added to the client and server models. This new application was written to produce traffic from trace files containing interarrival and response size information. Designated interarrival and response size filenames were given, and the client node reads times and sizes from them, scheduling requests to be made of the server. The client and servers shown in the testbed are configured as "mega-devices", that is, they have their packet forwarding rates and other settings adjusted so that, in the case of the server, they can accurately represent a large number of servers, or in the case of the client, the entire group of active DirecPC clients. (The presence of six servers in the testbed is intended to add a measure of interleaving to the packets flowing into the HGW.)

5.2 Modeling Studies on the HSTN Testbed

We seek to answer two questions about the models we have proposed for HSTNs. First, how well do the different distribution combinations work for performance prediction (and in particular, what is the "marginal utility" of the heavy-tailed distributions)? And second, once we have "trained" (meaning fit) our traffic model on our most recent data, how long can the model be expected to remain valid before the statistical characteristics of the arriving traffic change too much. The first question is one broadly concerned with which model (that is, combination of interarrival and response size distributions) is best suited to HSTN traffic, and whether or not heavy-tailed distributions provide significantly more accurate performance predictions. The second asks what the expected lifetime of a fitted model might be, when being applied to predicting traffic intensity or system performance for dynamic resource provisioning or quality of service (QoS) prediction.

5.2.1 Test Method

To answer these questions we set up the following test. All combinations of interarrival and response size distributions were fit to the first 30 minutes of traces 1, 2 and 3, exactly as detailed in Chapter 4, and synthetic traffic traces were generated for each model. For each trace, the succeeding 2, 5 and 10 minutes of traffic were run on the testbed, and separately 2, 5 and 10 minutes of each matching model's synthetic traffic, as well. For all runs a number of performance metrics were collected, to assess how well each model predicted traffic behavior.

5.2.2 Performance Metrics

Evaluating the predictive performance of traffic models requires well chosen metrics for comparison. The following metrics were collected for all runs:

- Peak throughput (in per-second intervals) on the satellite link – this allows us to gauge how well a model captures traffic burstiness;
- Average throughput on the satellite link – this is an excellent gauge of a model's overall prediction of traffic intensity;

- Maximum combined queue length on the SGW – this is an excellent example of a metric that would require accurate prediction for QoS provisioning;
- Average combined queue length on the SGW – another measure of overall traffic intensity;
- Peak number of simultaneous connections on the HGW – an important resource whose demand is desirable to predict;
- Peak delivery delay of UDP "probe" packets sent from an IS to the HH (at 200ms intervals);

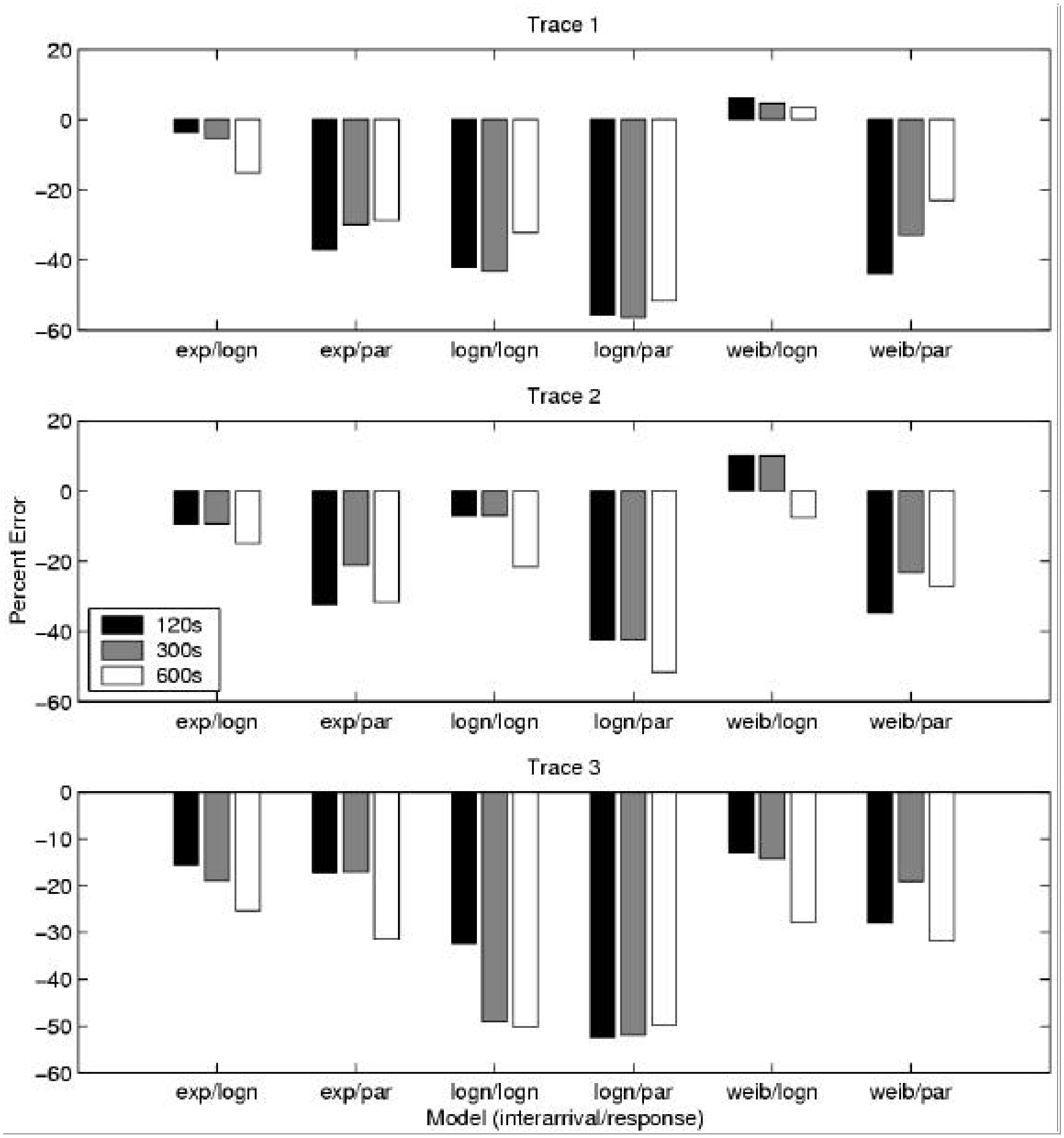

Figure 5.2 - Peak Throughput, Predicted vs. Actual (bytes/s)

		Interarrival and Response size distributions						
Trace	Future	Actual Trace	Exponential	Exponential	Log-normal	Log-normal	Weibull	Weibull
	time	Actual Trace	Log-normal	Pareto	Log-normal	Pareto	Log-normal	Pareto
1	120s	5161200	4971840	3248984	2984992	2293712	5492160	2895440
	300s	5250272	4971840	3679184	2984992	2293712	5492160	3524256
	600s	5851256	4976080	4175616	3970296	2839512	6058848	4509336
2	120s	4335824	3929656	2935816	4035072	2494128	4770256	2838384
	300s	4335824	3929656	3418248	4035072	2494128	4770256	3329992
	600s	5154616	4381456	3524136	4035072	2494128	4770256	3755312
3	120s	4495888	3788432	3717520	3043048	2133648	3916656	3239672
	300s	5976744	4841200	4955208	3043048	2872400	5119736	4835760
	600s	7233208	5400296	4955208	3598376	3635872	5227392	4925448

Table 5.1 - Peak Throughput, Predicted vs. Actual (bytes/s)

Table 5.1 tabulates the predicted vs. actual peak throughput for all six models fit to the first 30 minutes all three traces over the succeeding 120s, 300s, and 600s. Figure 5.2 shows the associated prediction error. The exponential interarrivals with log-normal download sizes model, and the Weibull/log-normal model perform best on traces 1 and 2. All others are poor predictors. No model performs very well for trace 3, but the exponential/log-normal and Weibull/log-normal are the best.

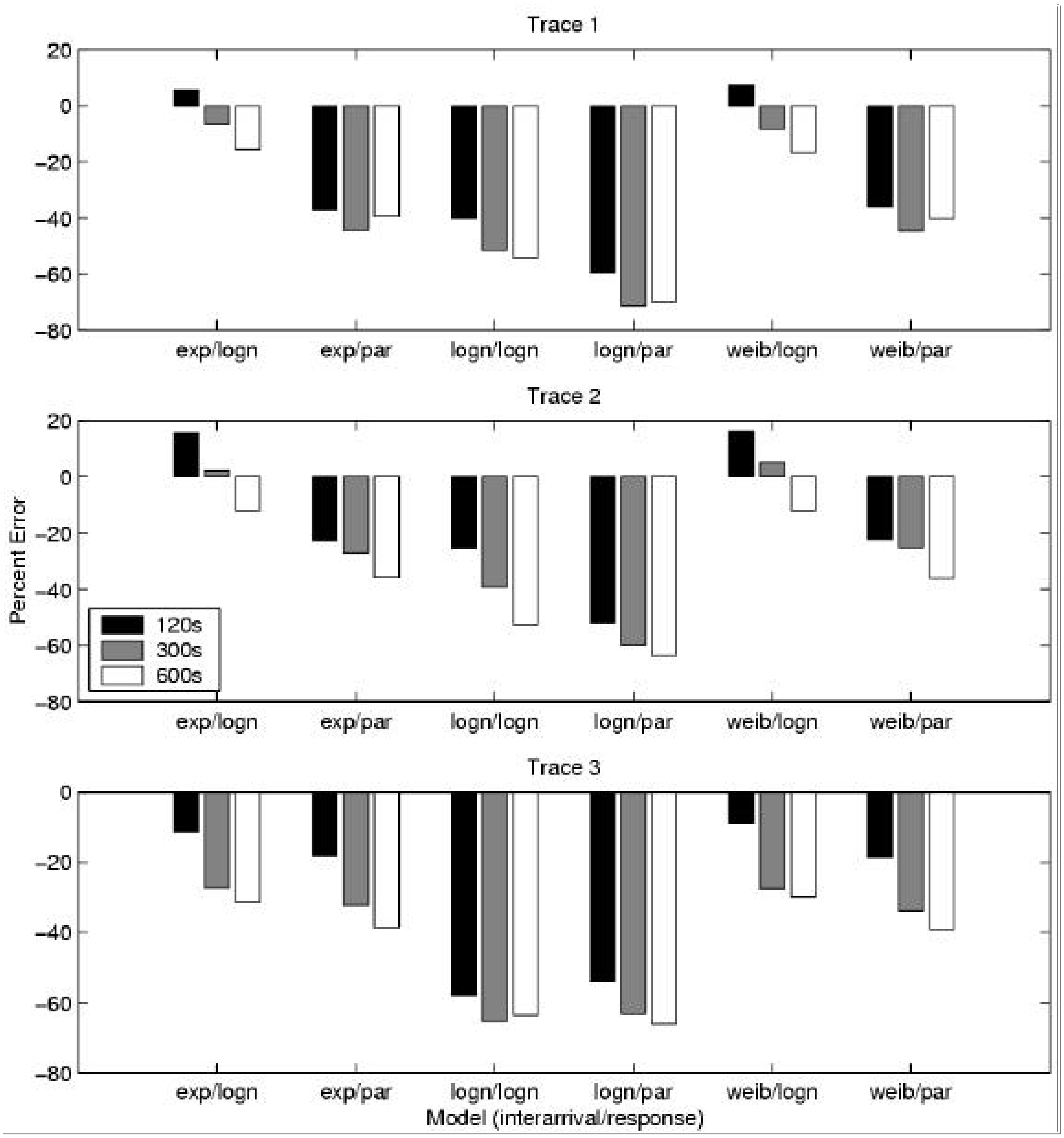

Figure 5.3 - Average Throughput, Predicted vs. Actual (bytes/s)

		Interarrival and Response size distributions						
Trace	Future time	Actual Trace Actual Trace	Exponential Log-normal	Exponential Pareto	Log-normal Log-normal	Log-normal Pareto	Weibull Log-normal	Weibull Pareto
1	120s	2388730	2525315	1503101	1431301	966787	2565716	1525621
	300s	2985151	2789657	1659555	1449717	860187	2739489	1651473
	600s	3344227	2822733	2024304	1532927	1015692	2785658	2001111
2	120s	1991141	2300186	1540868	1489039	956515	2313420	1549943
	300s	2219963	2270101	1617832	1346509	889262	2334632	1665257
	600s	2664985	2345479	1713316	1264444	967744	2344393	1705973
3	120s	2239213	1983998	1830608	944116	1032866	2041111	1820233
	300s	3308480	2401022	2245432	1146214	1218284	2395192	2187523
	600s	3816379	2615161	2348372	1391122	1297581	2678285	2328288

Table 5.2 - Average Throughput, Predicted vs. Actual (bytes/s)

Table 5.2 tabulates the predicted vs. actual average throughput for all six models. Figure 5.3 shows the associated prediction error. The exponential/log-normal and Weibull/log-normal models again perform best on all three traces. All others are poor predictors. Trace 3 is again the most difficult to predict, but for this metric the best case error is not as bad as in the peak throughput case. And, as in the peak throughput case, there is a general trend toward underestimation of the metric.

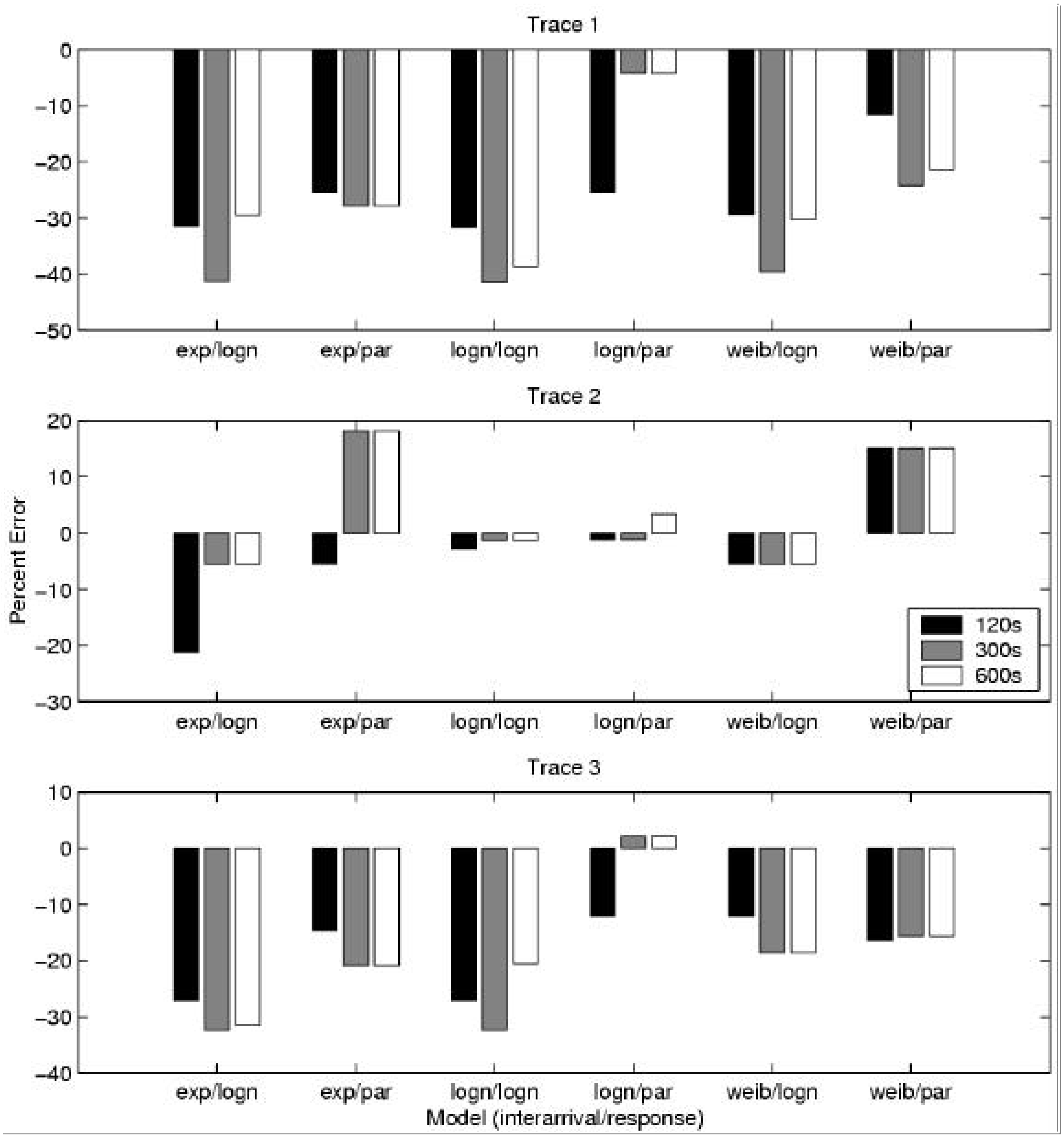

Figure 5.4 - Maximum SGW Queue Length, Predicted vs. Actual (bytes)

		Interarrival and Response size distributions						
Trace	Future	Actual Trace	Exponential	Exponential	Log-normal	Log-normal	Weibull	Weibull
	time	Actual Trace	Log-normal	Pareto	Log-normal	Pareto	Log-normal	Pareto
1	120s	4500	3088	3356	3076	3356	3179	3981
	300s	5255	3088	3795	3076	5032	3179	3981
	600s	5255	3703	3795	3220	5032	3667	4133
2	120s	3216	2534	3040	3128	3180	3040	3701
	300s	3216	3040	3799	3176	3180	3040	3701
	600s	3216	3040	3799	3176	3326	3040	3701
3	120s	3216	2534	3040	3128	3180	3040	3701
	300s	3216	3040	3799	3176	3180	3040	3701
	600s	3216	3040	3799	3176	3326	3040	3701

Table 5.3 - Maximum SGW Queue Length, Predicted vs. Actual (bytes)

Table 5.3 tabulates the predicted vs. actual maximum SGW queue length for all six models. Figure 5.4 shows the associated prediction error. Only one model performs adequately in predicting this metric: the Weibull/Pareto. All others are poor predictors. As with the previous metrics, there is a general trend toward underestimation.

.

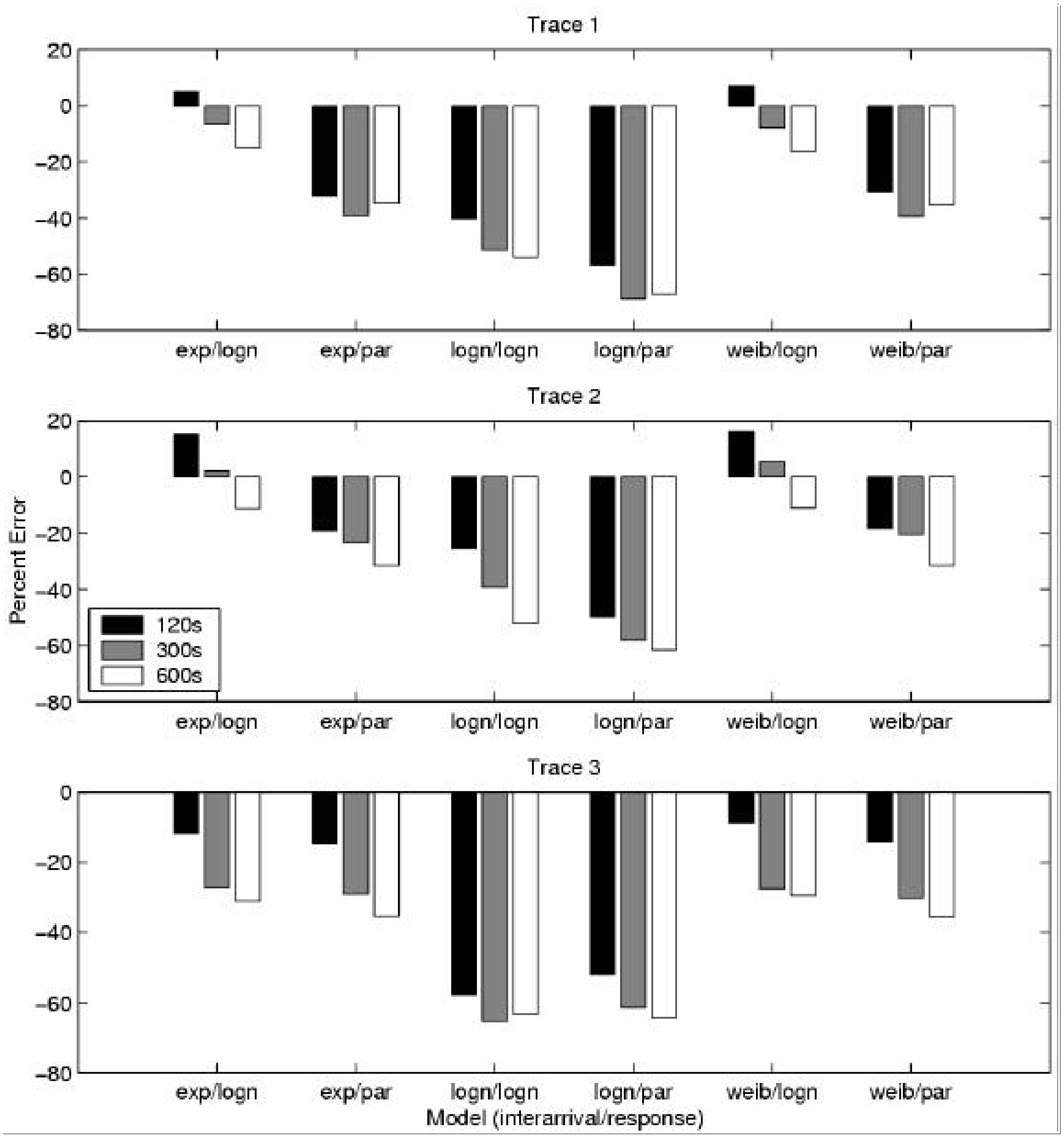

Figure 5.5 - Average SGW Queue Length, Predicted vs. Actual (bytes)

		Interarrival and Response size distributions						
Trace	Future time	Actual Trace Actual Trace	Exponential Log-normal	Exponential Pareto	Log-normal Log-normal	Log-normal Pareto	Weibull Log-normal	Weibull Pareto
1	120s	6.46	6.79	4.39	3.86	2.79	6.92	4.47
	300s	8.03	7.50	4.88	3.90	2.52	7.40	4.85
	600s	8.95	7.61	5.86	4.12	2.96	7.51	5.79
2	120s	5.34	6.15	4.32	3.99	2.68	6.21	4.36
	300s	5.96	6.09	4.57	3.62	2.49	6.27	4.74
	600s	7.09	6.30	4.85	3.39	2.72	6.31	4.85
3	120s	6.03	5.32	5.14	2.55	2.90	5.51	5.17
	300s	8.84	6.42	6.27	3.07	3.41	6.40	6.17
	600s	10.16	7.00	6.56	3.71	3.64	7.16	6.54

Table 5.4 - Average SGW Queue Length, Predictions vs. Actual (bytes)

Table 5.4 tabulates the predicted vs. actual average SGW queue length for all six models. Figure 5.5 shows the associated prediction error. The exponential/log-normal and Weibull/log-normal models again perform best on all three traces, but as in the throughput metrics, do not perform particularly well for trace 3. All others are poor predictors. And, as in the peak throughput case, there is a general trend toward underestimation of the metric, particularly for longer prediction times.

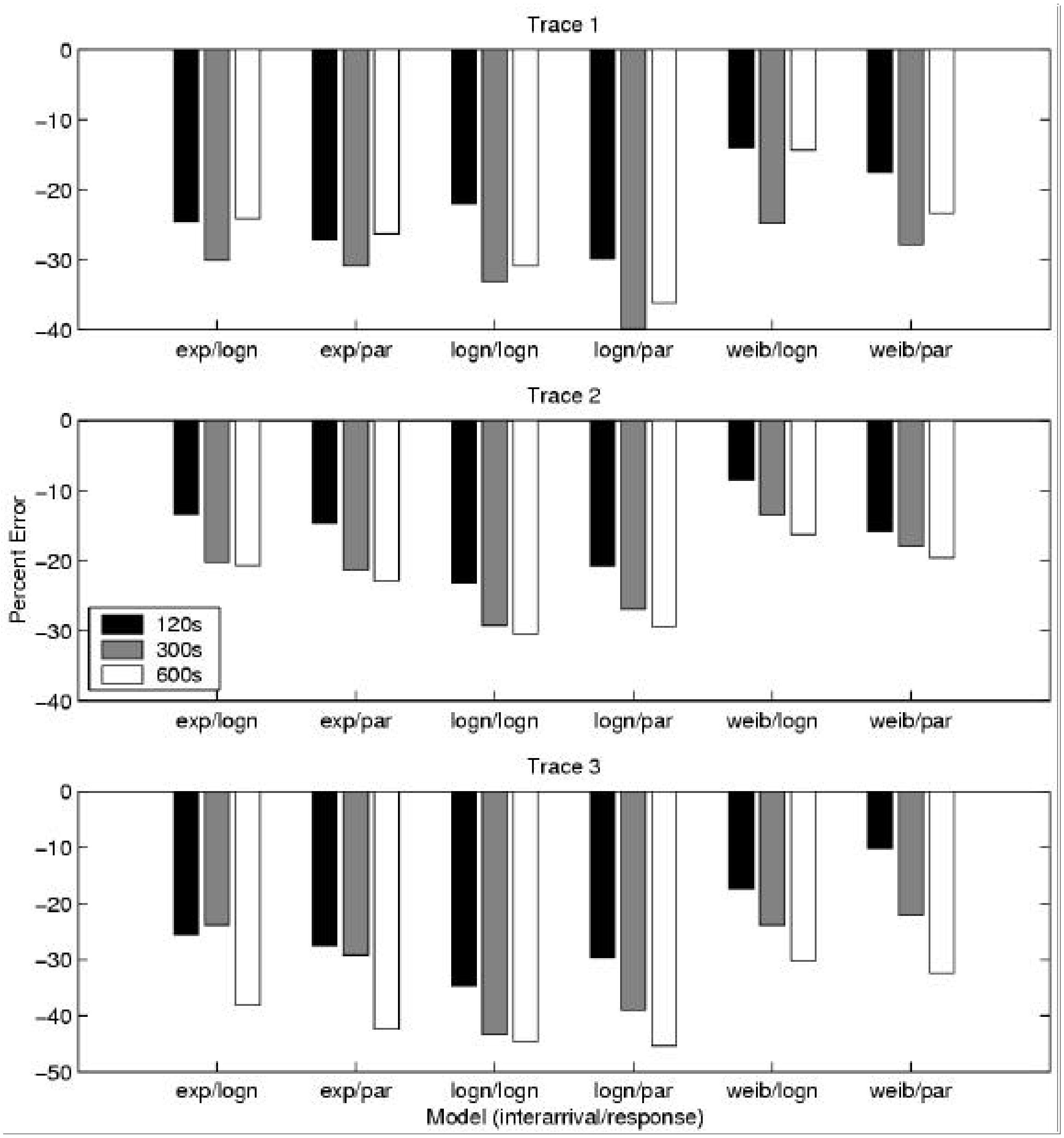

Figure 5.6 - Peak Number of Connections on HGW, Predicted vs. Actual

		Interarrival and Response size distributions						
Trace	Future time	Actual Trace	Exponential	Exponential	Log-normal	Log-normal	Weibull	Weibull
		Actual Trace	Log-normal	Pareto	Log-normal	Pareto	Log-normal	Pareto
1	120s	114	86	83	89	80	98	94
	300s	133	93	92	89	80	100	96
	600s	133	101	98	92	85	114	102
2	120s	82	71	70	63	65	75	69
	300s	89	71	70	63	65	77	73
	600s	92	73	71	64	65	77	74
3	120s	98	73	71	64	69	81	88
	300s	113	86	80	64	69	86	88
	600s	139	86	80	77	76	97	94

Table 5.5 – Peak Number of Spoofed Connections, Predicted vs. Actual

Table 5.5 tabulates the predicted vs. actual peak number of HGW connections for all six models. Figure 5.6 shows the associated prediction error. Weibull/log-normal model performs marginally well on trace 1 and 2. All others are poor predictors. Predictions for this metric for trace 3 are inadequate for all models. There is a strong trend toward underestimation of this metric.

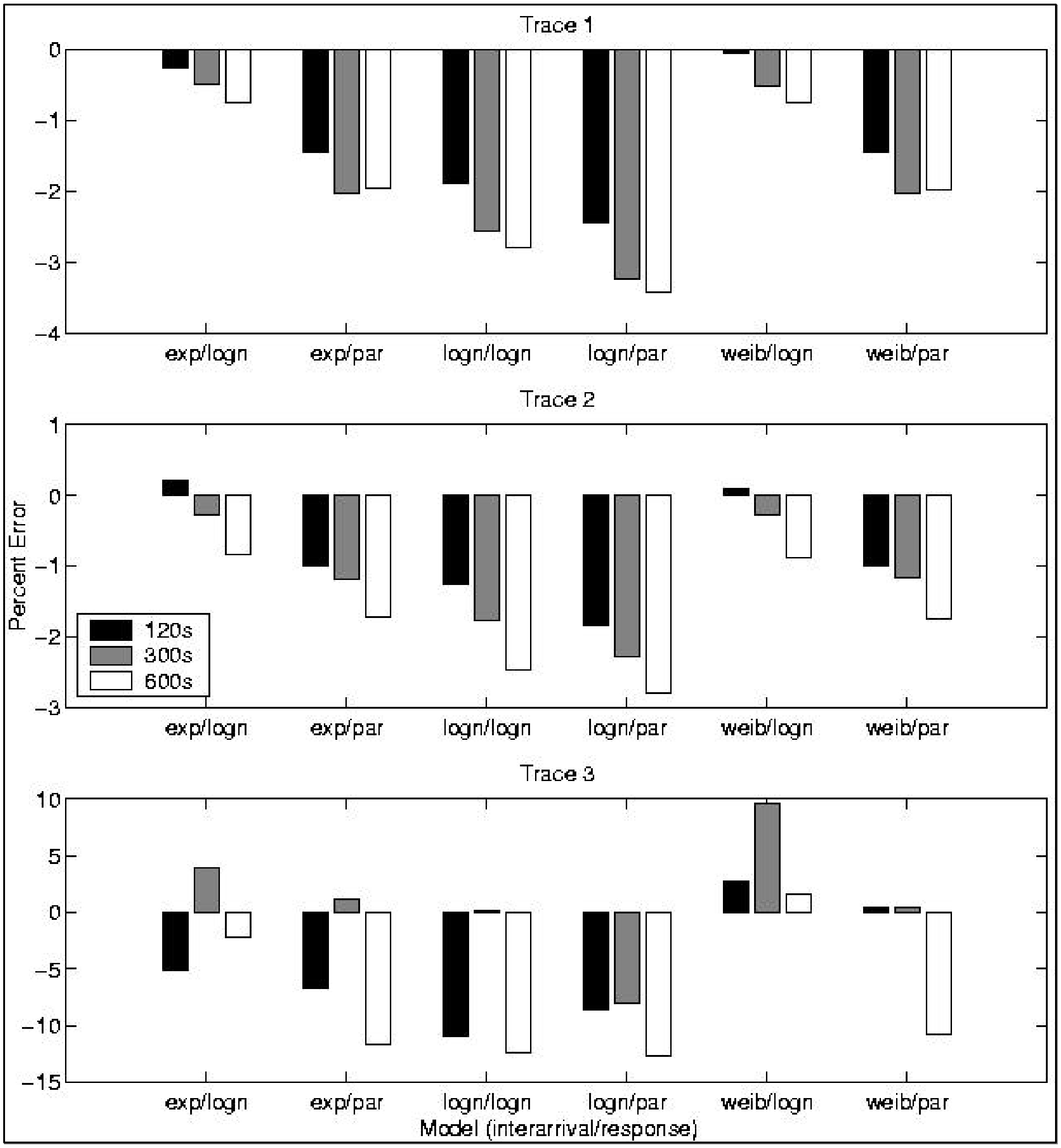

Figure 5.7 - Average UDP Packet Delivery Delay, Predicted vs. Actual (sec)

		Interarrival and Response size distributions						
Trace	Future time	Actual Trace Actual Trace	Exponential Log-normal	Exponential Pareto	Log-normal Log-normal	Log-normal Pareto	Weibull Log-normal	Weibull Pareto
1	120s	0.312435	0.311659	0.307931	0.306541	0.304823	0.312294	0.307949
	300s	0.314506	0.312997	0.308146	0.306415	0.304359	0.312893	0.308166
	600s	0.315622	0.313259	0.309476	0.306783	0.304783	0.313267	0.309404
2	120s	0.309643	0.310292	0.306581	0.305746	0.303934	0.309905	0.30656
	300s	0.310775	0.309921	0.307054	0.305304	0.3037	0.309892	0.307153
	600s	0.312754	0.31014	0.307397	0.305026	0.304025	0.309975	0.307272
3	120s	0.361799	0.3432	0.33742	0.32226	0.330512	0.37181	0.363305
	300s	0.361799	0.375965	0.365902	0.362493	0.332846	0.396664	0.363305
	600s	0.414312	0.405213	0.365902	0.462799	0.362113	0.420792	0.369727

Table 5.6 – Average UDP Packet Delivery Delay, Predicted vs. Actual (sec)

Table 5.6 tabulates the predicted vs. actual UDP packet delivery delay for all six models. Figure 5.7 shows the associated prediction error. The exponential/log-normal and Weibull/log-normal models again perform best on all three traces. All others are poor predictors. As with most of the other metrics, there is a general trend toward underestimation of the metric.

These results are quite interesting, though in some respects difficult to interpret. Two models seem to generally do the best job: the Weibull interarrivals with log-normal response sizes and the exponential interarrivals with log-normal response sizes. The performance of the first model is not surprising, since we have already established that the Weibull and log-normal distributions are the ones best fitting interarrivals and responses, respectively.

What is more surprising is that the exponential/log-normal model performs almost as well as the Weibull/log-normal one. This implies that for accurate performance prediction, a good response size distribution fit is more critical than the interarrival distribution fit. In our studies we have done performance prediction through simulation—making any distribution equally easy to use. However, if performance predictions are to be obtained through analytical means the use of exponential interarrivals may simplify the analysis and therefore the simplicity/accuracy tradeoff is one worth considering.

For all metrics the log-normal interarrival distribution performs more poorly. This is likely due, at least in part, to the fact that the heavier-tailed log-normal slightly underestimates connection arrival intensity (which is visible in the error graphs).

Log-normal appears to generally be the preferable response size distribution match. But on one metric, the maximum SGW queue length, it is outperformed by the Pareto, for all interarrival distribution pairings, on two of the three traces.

There appears to be a general trend among the models of underestimating the metrics. Even the best fitting models, the exponential/log-normal and Weibull/log-normal combinations, generally give low predictions of the performance parameters we chose. This could in part be due to an increase in traffic intensity over the succeeding 10 minutes in the actual traces (this is particularly possible for Trace 3, where, upon review, the mean interarrival time of the succeeding 10 minutes was noticeably smaller than for the 30 minutes to which the models were fit). But it also appears that even our best models simply fail to fully capture the burstiness of HSTN traffic. This conclusion is supported by the fact that performance prediction is worse for the "peak" or maximum metrics than for the average metrics.

Most difficult to ascertain is how quickly the models' fitnesses grow stale. There does appear to be sufficient evidence to conclude that this is occurring at time scales as small as 5 to 10 minutes. It is most evident in the averaging statistics, because the unavoidable error in predicting maxima over time scales as small as 2 minutes tends to mask the phenomenon in the "peak" type statistics.

Chapter 6 - Conclusions

6.1 Conclusions

Performance prediction via fitted traffic models is a tricky task, magnified in difficulty in HSTNs by the requirement of connection level models. Capturing the burstiness, or self-similarity, of traffic is essential for accurate performance prediction. Heavy-tailed distributions, like the Weibull, log-normal and Pareto (particularly when applied to response sizes), do provide a higher degree of burstiness, but still fall short for some metrics.

It also appears that models—once fit—grow stale relatively quickly. This presents a predicament. To accurately fit a model we must include a sufficient number of samples. In the case of our DirecPC™ traces, connections arrived at a rate on the order of 1,000 per minute. Given that heavy-tailed data (such as the download sizes) requires an especially large number of samples to provide an accurate fit, 10 minutes would seem to be the minimum amount of time over which to fit our model. However our results show that the underlying traffic process may barely be stationary over this amount of time. This is an empirical manifestation of the fact that no theory for non-linear prediction has been developed for the statistical models (except for the case of Gaussian self-similar processes).

6.2 Future Work

This thesis did not address analytical performance prediction, choosing instead to predict performance by simulation. A similar study of the performance predictive utility of these traffic models, but featuring analytical results, would be equally enlightening.

Other interesting expansions on the work presented here would include varying the amount of “past” time the models are fit to, and further increasing the amount of “future” time they are used to predict. This study has only highlighted the limited lifetime of a fit model.

In addition, it might also be possible to “weight” the trace data so that the more recent past figures more prominently in the calculation of model fits (the idea being to “ground” our models in a significant amount of historical data, but still make them flexible enough to accommodate recent changes in the traffic characteristics).

Bibliography

[BU] Boston University web traces, ftp://cs-ftp.bu.edu/techreports/95-010-www-client-traces.tar.gz.

[Bux81] W. Bux, “Local-Area Subnetworks: A Performance Comparison,” *IEEE Transactions on Communications*, vol. 29, no. 10, pp. 1465-73, October 1981.

[BSTW94] J. Beran, R. Sherman, M. Taqqu, W. Willinger, “Long-Range Dependence in Variable-Bit-Rate Video Traffic,” *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 1566-78, February/March/April 1995.

[CBC95] C. Cunha, A. Bestavros, M. Crovella, “Characteristics of WWW Client-based Traces,” Technical Report BU-CS-95-010, Computer Science Department, Boston University, July 1995.

[CM80] W. Cook, D. Mumme, “Estimation of Pareto Parameters by Numerical Methods,” in *Statistical Distributions for Scientific Work, Vol. 5*, ed. by Charles Taillie et al., D. Reidel Pub. Co., Boston, 1980, pp. 127-132.

[DMRW94] D. Duffy, A. McIntosh, M. Rosenstein, W. Willinger, “Statistical Analysis of CCSN/SS7 Traffic Data from Working Subnetworks,” *IEEE Journal of Selected Areas in Communications*, vol. 12, no. 3, pp. 544-51, 1994.

[F92] P. Flandrin, “Wavelet Analysis and Synthesis of Fractional Brownian Motion,” *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 910-17, March 1992.

[Fag75] M. Fagen, Ed., *A History of Engineering and Science in the Bell System, The Early Years (1875-1925)*, Bell Telephone Laboratories, Incorporated, 1975, ISBN 0608080667.

[Feld95] A. Feldman, *Online Call Admission for High-Speed Networks*, Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, October 1995, Technical Report CMU-CS-95-201.

[FL91] H. Fowler, W. Leland, “Local Area Network Traffic Characteristics, with Implications for Broadband Network Congestion Management,” *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 7, pp. 1139-49, September 1991.

[GW94] M. Garret, W. Willinger, “Analysis, Modeling and Generation of Self-Similar VBR Video Traffic,” in Proceedings of the ACM SIGCOMM ’94, London, pp. 269-80, 1994.

[Heff80] H. Heffes, "A Class of Data Traffic Processes—Covariance Function Characterization and Related Queueing Results," *Bell System Technical Journal*, vol. 59, no. 6, pp. 897-929, July-August 1980.

[HL86] H. Heffes, D. Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance," *IEEE Journal on Selected Areas in Communications*, pp. 856-68, 1986.

[HLF98] F. Huebner, D. Liu, J. Fernandez, "Queueing Performance Comparison of Traffic Models for Internet Traffic," AT&T Labs technical report, http://www.att.com/technology/history/archive/1998/pdf/que.pdf, 1998.

[ITA] The Internet Traffic Archive, http://ita.ee.lbl.gov/index.html.

[Jac98] V. Jacobson, *tcpdump* program, version 3.4, http://www-nrg.ee.lbl.gov/nrg.html, 1998.

[JB88] V. Jacobson, R. Braden, "TCP Extensions for Long-Delay Paths," Internet RFC 1072, http://www.faqs.org/rfcs/rfc1072.html, October 1988.

[JR86] R. Jain, S. Routhier, "Packet Trains—Measurements and a New Model for Computer Network Traffic," *IEEE Journal on Selected Areas in Communications*, vol. SAC-4, no. 6, pp. 986-95, September 1986.

[KLBB99] M. Karir, M. Liu, B. Barrett, J. Baras, "A Simulation Study of Enhanced TCP/IP Gateways for Broadband Internet over Satellite," in *Proceedings of OPNETWORK'99*, 1999.

[LEWW95] W.-C. Lau, A. Erramilli, J. Wang, W. Willinger, "Self-Similar Traffic Generation: The Random Midpoint Displacement Algorithm and its Properties," *ICC*, pp. 466-72, 1995.

[LTWW94] W. Leland, M. Taqqu, W. Willinger, D. Wilson, "On the Self-Similar Nature of Ethernet Traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1-15, February 1994.

[LW91] W. Leland, D. Wilson, "High time-resolution measurement and analysis of LAN traffic: Implications for LAN Interconnection," in Proceedings of IEEE INFOCOM '91, Bal Harbour, FL, pp.1360-66, 1991.

[LKRB99] M. Liu, M. Karir, M. Raissi-Dehkordi, J. Baras, "Hybrid Internet Simulation Testbed," in *Proceedings of OPNETWORK'99*, and CSHCN Technical Report TR 99-23, http://www.isr.umd.edu/TechReports/CSHCN/1999/CSHCN_TR_99-23/CSHCN_TR_99-23.phtml, 1999.

[Ma63] B. Mandelbrot, "New Methods in Statistical Economics," *Journal of Polictical Economy*, vol. 71, no. 5, pp. 421-40, October 1963.

[MB76] R. Metcalfe, D. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," *Communications of the ACM*, vol. 19, pp. 395-404, 1976.

[Pax94] V. Paxson, "Empirically Derived Analytic Models of Wide-Area TCP Connections", *IEEE/ACM Transactions on Networking*, vol. 2, pp.316-336, August 1994.

[PF94] V. Paxson, S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," in Proceedings of the ACM SIGCOMM '94, London, pp. 257-68, August 1994.

[Pax95] V. Paxson, *tcp-reduce* script, version 1.0, from the *Internet Traffic Archive*, http://ita.ee.lbl.gov/html/contrib/tcp-reduce.html, 1995.

[Pax97] V. Paxson, "Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic," *Computer Communication Review*, vol. 27, no. 5, pp. 5-18, October 1997.

[PJ90] S. Pederson, M. Johnson, "Estimating Model Discrepancy," *Technometrics*, vol. 32, no. 3, pp 305-14.

[Stal98] W. Stallings, *High-Speed Networks, TCP/IP and ATM Design Principles*, Prentice-Hall, Inc., 1998, ISBN 0135259657.

[WP98] W. Willinger, V. Paxson, "Where Mathematics meets the Internet," *Notices of the American Mathematical Society*, vol. 45, no. 8, pp. 961-70, September 1998.

[WPT98] W. Willinger, V. Paxson, M. Taqqu, "Self-Similarity and Heavy Tails: Structural Modeling of Network Traffic," from the book *A Practial Guide to Heavy Tails: Statisitcal Techniques and Applications*, pp. 27-53, R. Adler, R. Feldman, M. Taqqu, editors, Birkhauser, Boston 1998 ISBN 0817639519.

[WTSW97] W. Willinger, M. Taqqu, R. Sherman, D. Wilson, "Self-Similarity Through High Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level," *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, pp. 71-86, 1997.